



UNIVERSIDADE FEDERAL DE SERGIPE  
CENTRO DE CIÊNCIAS EXATAS E TECNOLOGIA  
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

## **Segmentação e Reconhecimento de Semáforos com Ajuda de Instrução Especialista**

Dissertação de Mestrado

Thiago da Silva Almeida



São Cristóvão – Sergipe

2018

UNIVERSIDADE FEDERAL DE SERGIPE  
CENTRO DE CIÊNCIAS EXATAS E TECNOLOGIA  
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

Thiago da Silva Almeida

## **Segmentação e Reconhecimento de Semáforos com Ajuda de Instrução Especialista**

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Ciência da Computação da Universidade Federal de Sergipe como requisito parcial para a obtenção do título de mestre em Ciência da Computação.

Orientador(a): Dr. Hendrik Teixeira Macedo  
Coorientador(a): Dr. Leonardo Nogueira Matos

São Cristóvão – Sergipe

2018

**FICHA CATALOGRÁFICA ELABORADA PELA BIBLIOTECA CENTRAL  
UNIVERSIDADE FEDERAL DE SERGIPE**

Almeida, Thiago da Silva  
S447s      Segmentação e reconhecimento de semáforos com ajuda de  
instrução especialista / Thiago da Silva Almeida ; orientador  
Hendrik Teixeira Macedo . - São Cristóvão, 2018.  
57 f. : il.

Dissertação (mestrado em Ciência da Computação) –  
Universidade Federal de Sergipe, 2018.

1. Computação. 2. Aprendizado do computador. 3. Aquisição  
de conhecimento (Sistemas especialistas). 4. Trânsito – sinais e  
sinalização. 5. Inteligência artificial I. Macedo, Hendrik Teixeira  
orient. II. Título.

CDU 004:656.056.4





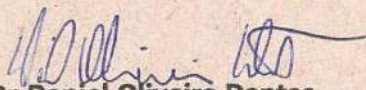
**UNIVERSIDADE FEDERAL DE SERGIPE**  
**PRÓ-REITORIA DE PÓS-GRADUAÇÃO E PESQUISA**  
**COORDENAÇÃO DE PÓS-GRADUAÇÃO**  
**PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO**


**Ata da Sessão Solene de Defesa da Dissertação do**  
**Curso de Mestrado em Ciência da Computação-UFS.**  
**Candidato: THIAGO DA SILVA ALMEIDA**

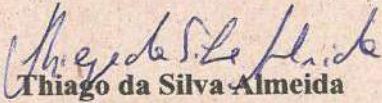
Em 26 dias do mês de outubro do ano de dois mil e dezoito, com início às 09h00min, realizou-se na Sala de Seminários do DCOMP da Universidade Federal de Sergipe, na Cidade Universitária Prof. José Aloísio de Campos, a Sessão Pública de Defesa de Dissertação de Mestrado do candidato **Thiago da Silva Almeida**, que desenvolveu o trabalho intitulado: "*Segmentação e Reconhecimento de Semáforos com Ajuda de Instrução Especialista*", sob a orientação do Prof. Dr. **HENDRIK TEIXEIRA MACEDO**. A Sessão foi presidida pelo Prof. Dr. **HENDRIK TEIXEIRA MACEDO** (PROCC/UFS), que após a apresentação da dissertação passou a palavra aos outros membros da Banca Examinadora, Prof. Dr. **DANIEL OLIVEIRA DANTAS** (PROCC/UFS) e, em seguida, ao Prof. Dr. **PAULO SALGADO GOMES DE MATTOS NETO** (UFPE). Após as discussões, a Banca Examinadora reuniu-se e considerou o mestrando (a) APROVADO "(aprovado/reprovado)" COM "(com/sem)" ressalvas. Atendidas as exigências da Instrução Normativa 01/2017/PROCC, do Regimento Interno do PROCC (Resolução 67/2014/CONEPE), e da Resolução nº 25/2014/CONEPE que regulamentam a Apresentação e Defesa de Dissertação, e nada mais havendo a tratar, a Banca Examinadora elaborou esta Ata que será assinada pelos seus membros e pelo mestrando.

Cidade Universitária "Prof. José Aloísio de Campos", 26 de outubro de 2018.

  
**Prof. Dr. Hendrik Teixeira Macedo**  
**(PROCC/UFS)**  
**Presidente**

  
**Prof. Dr. Daniel Oliveira Dantas**  
**(PROCC/UFS)**  
**Examinador Interno**

  
**Prof. Dr. Paulo Salgado Gomes de Mattos Neto**  
**(UFPE)**  
**Examinador Externo**

  
**Thiago da Silva Almeida**  
**Candidato**



*Só acredito no semáforo*  
*Só acredito no avião*  
*Eu acredito no relógio*  
*Acredito no coração*  
*(Vanguard)*

# Resumo

Impulsionadas pelo alto interesse no desenvolvimento de veículos autônomos, as pesquisas sobre detecção e reconhecimento de semáforo tem crescido a cada ano. Além disso, técnicas de Aprendizado de Máquina tem sido amplamente utilizadas não apenas na pesquisa sobre reconhecimento de semáforos, como também em todos os campos em que pode ser necessário encontrar padrões em dados e automatizar um comportamento humano. Muitos dos algoritmos de Aprendizado de Máquina mais utilizados requerem uma grande quantidade de dados para funcionar adequadamente e, por causa disso, precisam de muito poder computacional. Nesta dissertação foi investigado o uso de Conhecimento ou Instrução Especialista - IE na tentativa de reduzir a quantidade de dados exigidos por um algoritmo de Aprendizado de Máquina. O conhecimento especialista escolhido para ser utilizado como instrução foi a localização do semáforo numa dada imagem obtida do interior do veículo. Esta estratégia se baseia na teoria de que há maior probabilidade do semáforo aparecer em certas regiões como, por exemplo, nas regiões centrais e superiores da imagem. Foram construídos mapas de frequência da localização de semáforos para validar essa teoria. Os mapas são o resultado da análise de um especialista humano sobre uma série de imagens, o especialista marca em cada imagem as coordenadas da região em que o(s) semáforo(s) aparece(m). Os resultados mostram que a inclusão do IE aumentou em pelo menos 15% a acurácia obtida pelo algoritmo de classificação em dois bancos de imagem diferentes. As taxas de avaliação alcançadas pela inclusão de IE também foram superiores no experimento completo incluindo a detecção de semáforo seguida da classificação pelo algoritmo treinado. Neste teste a inclusão de IE na PCANet obteve 83% de precisão e 73% de cobertura, enquanto a PCANet sem inclusão de IE alcançou 75,3% de precisão e 51,1% de cobertura. Este trabalho também apresenta um protótipo de Dispositivo Reconhecedor de Semáforos (DRS) que pode ser utilizado para criar bancos de imagens e como suporte ao motorista. O DRS proposto utiliza um smartphone como câmera e unidade de processamento. Para validar o layout do protótipo, um conjunto de dados foi obtido e testado em um algoritmo de filtro de supressão de fundo adaptável (Adaptive Background Suppression Filter - AdaBSF) e Support Vector Machines (SVMs) para detectar e reconhecer semáforos. A aplicação do AdaBSF e subsequente classificação com SVM ao conjunto de dados alcançou uma taxa de precisão de 100% e reconvoção/cobertura de 65%.

**Palavras-chave:** Aprendizado de Máquina, Conhecimento Especialista, Detecção e Reconhecimento de Semáforos, Visão Computacional.

# Abstract

Traffic light detection and recognition research has grown every year. In addition to this, Machine Learning have been largely used not only in traffic light research but in every field that it is useful and possible to generalize data and automatize a human behavior. Machine Learning algorithms requires a large amount of data to work properly, and because of that, needs a lot of computational power to analyze the data. In this article Expert Knowledge was used in an attempt to reduce the amount of data required by a Machine Learning algorithm. Results show that the inclusion of Expert Knowledge - EK - increased in at least 15% the algorithm test accuracy rate in two different image datasets. The EK used was the traffic light location in a image obtained from a vehicle interior. This idea is based on the theory that there are regions where the traffic lights appear more frequently and, for that reason, those regions have a bigger traffic light appearance probability. Traffic light frequency maps were built to validate this theory. The maps are the result of a human expert analysis over a image group containing traffic images with traffic lights, the human expert tagged in each image the location where the traffic light appeared. The EK inclusion evaluation rates were also superior when testing the detection algorithm followed by the trained classification algorithm, in this test the EK inclusion obtained 83% precision rate and 73% recall rate, while the traditional trained algorithm had 75,3% precision rate and 51,1% recall rate. This article also proposes a traffic light recognition (TLR) device prototype using a smartphone as camera and processing unit that can be used as a driver assistance. To validate this layout prototype, a dataset was built and used to test an algorithm that uses an adaptive background suppression filter (AdaBSF) and Support Vector Machines (SVMs) to detect traffic lights. The application of AdaBSF and subsequent classification with SVM to the dataset achieved 100% precision rate and recall of 65%.

**Keywords:** Machine Learning, Expert Knowledge, Traffic Light Detection and Recognition, Computer Vision.

# Lista de ilustrações

Figura 1 – Estrutura funcional completa de um sistema de processamento de imagens. . .	15
Figura 2 – Imagem digital e respectiva representação matricial. . . . .	19
Figura 3 – Exemplo de RNA multicamadas típico. . . . .	21
Figura 4 – Exemplo de arquitetura de uma CNN. . . . .	22
Figura 5 – Ilustração de um hiperplano ótimo encontrado por uma SVM. . . . .	23
Figura 6 – Mapa de frequência de semáforos obtido a partir da amostra aleatória do dataset criado por Shi, Zou e Zhang (2016) . . . . .	32
Figura 7 – Mapa de frequência de semáforos obtido a partir da amostra aleatória do dataset criado por Chen e Huang (2016) . . . . .	33
Figura 8 – Histograma da distribuição das frequências do mapa da Figura 6 . . . . .	33
Figura 9 – Histograma da distribuição das frequências do mapa da Figura 7 . . . . .	34
Figura 10 – Estrutura de uma PCANet de duas fases (Stages) com inclusão de IE. . . .	36
Figura 11 – Acurácias de teste com o banco de Shi, Zou e Zhang (2016) após o treinamento com IE (linha laranja) e sem IE (linha azul) . . . . .	38
Figura 12 – Acurácias de teste com o banco de Chen e Huang (2016) após o treinamento com IE (linha laranja) e sem IE (linha azul) . . . . .	39
Figura 13 – Acurácias de teste do grupo de dados GREENROI1 do dataset de Chen e Huang (2016). após o treinamento com IE (linha laranja) e sem IE (linha azul)	40
Figura 14 – Acurácias de teste do grupo de dados GREENROI3 do dataset de Chen e Huang (2016). após o treinamento com IE (linha laranja) e sem IE (linha azul)	40
Figura 15 – Acurácias de teste do grupo de dados GREENROI4 do dataset de Chen e Huang (2016). após o treinamento com IE (linha laranja) e sem IE (linha azul)	41
Figura 16 – Acurácias de teste do grupo de dados REDROI1 do dataset de Chen e Huang (2016). após o treinamento com IE (linha laranja) e sem IE (linha azul) . . .	41
Figura 17 – Acurácias de teste do grupo de dados REDROI3 do dataset de Chen e Huang (2016). após o treinamento com IE (linha laranja) e sem IE (linha azul) . . .	42
Figura 18 – iPhone 6 em suporte para dispositivo DRS. . . . .	46
Figura 19 – Imagens obtidas utilizando o suporte para DRS prototipado com diferentes smartphones. De cima para baixo, em sentido horário: imagem obtida com um Motorola G 2 <sup>a</sup> Geração, com um iPhone 6, e com um Galaxy S8+. . . .	47
Figura 20 – Taxas de precisão e cobertura ( <i>recall</i> ) por smartphone utilizado para capturar as imagens. . . . .	48



Figura 21 – Da esquerda para a direita: exemplos de semáforo verde e vermelho utilizados no treinamento, exemplos de semáforo verde e vermelho obtidos pelo DRS prototipado e utilizados nos testes. . . . .	48
Figura 22 – Taxas de precisão e cobertura ( <i>recall</i> ): da nossa reprodução do trabalho de Shi, Zou e Zhang (2016), do trabalho original de Shi, Zou e Zhang (2016), e dos testes utilizando as imagens obtidas pelo DRS prototipado, respectivamente.	49

# Lista de abreviaturas e siglas

2D	Bidimensional
3D	Tridimensional
AM	Aprendizado de Máquina
CMYK	Cyan, Magenta, Yellow, Key
CNN	Convolutional Neural Network
DRS	Dispositivo Reconhecedor de Semáforos
IE	Instrução Especialista
GPS	Global Positioning System
HMM	Hidden Markov Models
HSV	Hue, Saturation, Value
HSB	Hue, Saturation, Brightness
PCANet	Principal Component Analysis Network
PI	Processamento de Imagens
RGB	Red, Green, Blue
RN	Redes Neurais
RNA	Redes Neurais Artificiais
ROI	Regions Of Interest
SVM	Support Vector Machine
TAE	Teoria do Aprendizado Estatístico
TLR	Traffic Light Recognition
VC	Visão Computacional
XP	Extreming Programming

# Sumário

<b>1</b>	<b>Introdução</b>	<b>10</b>
1.1	Hipótese	12
1.2	Objetivos	12
1.2.1	Objetivo Geral	12
1.2.2	Objetivos Específicos	12
<b>2</b>	<b>Referencial Teórico</b>	<b>14</b>
2.1	Processamento de Imagens e Visão Computacional	14
2.1.1	Passos Fundamentais em Processamento de Imagens	15
2.1.2	Fundamentos de Imagens Digitais	18
2.2	Aprendizado de Máquina	19
2.2.1	Redes Neurais Artificiais	20
2.2.2	<i>Support Vector Machines</i>	22
2.3	Revisão Bibliográfica	23
2.3.1	Datasets	27
<b>3</b>	<b>Aprendizado de Máquina com Instrução Especialista</b>	<b>28</b>
3.1	Metodologia de Avaliação	30
3.1.1	Experimento 1	30
3.1.2	Experimento 2	30
3.2	Instrução Especialista	31
3.3	Treinamento com Instrução Especialista	34
3.4	Deteção e Reconhecimento Especialista	39
<b>4</b>	<b>Protótipo de um Dispositivo de Deteção e Reconhecimento de Semáforos</b>	<b>45</b>
4.1	Resultados Obtidos com o Protótipo	46
<b>5</b>	<b>Conclusão</b>	<b>50</b>
	<b>Referências</b>	<b>52</b>

# 1

## Introdução

Em 2015, segundo dados da Polícia Rodoviária Federal, dos mais de 150 mil acidentes registrados, cerca de 7,5 mil foram relacionados à desobediência à sinalização de trânsito. Das 7,5 mil vítimas fatais, 425 morreram por ter cometido a infração (Gonçalves, Jéssica., 2015).

Observando o estudo publicado pela CET - Companhia de Engenharia de Tráfego (2016), percebe-se que em agosto de 2016 o avanço do sinal vermelho foi a segunda infração que mais apareceu associada a acidentes fatais envolvendo motocicletas, carros e ônibus na cidade de São Paulo.

Embora o semáforo seja utilizado amplamente no mundo inteiro como dispositivo regulador do tráfego e seja de entendimento muito simples e lógico, os números citados provam que acidentes são comuns e com consequências graves como a morte de condutores, passageiros e pedestres. Algumas situações em especial podem ser citadas como prováveis causas do avanço do sinal vermelho:

- Semáforos mal localizados que não podem ser visualizados a tempo, ou ambíguos;
- Semáforos com defeito, desligado, ou com luz muito fraca;
- Iluminação do ambiente que atrapalha a visão do condutor; (JENSEN et al., 2016)
- Deficiência visual do condutor;
- Dúvida se há tempo suficiente para cruzar a via quando o sinal fica amarelo; (JENSEN et al., 2016)
- Quantidade de itens reguladores do trânsito a serem observados.

Os dois primeiros itens citados podem ser facilmente resolvidos com a dedicação do órgão regulador de trânsito em dispor e manter os semáforos de forma otimizada nas ruas.

Entretanto, a problemática apresentada pelos itens restantes pode ser minimizada com o uso de um Dispositivo Reconhecedor de Semáforos - DRS.

De acordo com Jensen et al. (2016), a principal tarefa de um DRS é evitar acidentes e salvar vidas por informar ao condutor da existência de um semáforo vermelho ou amarelo à frente de forma não-intrusiva ao motorista, utilizando talvez um aviso sonoro. Além disso, um DRS pode trazer outras informações como qual o principal semáforo para aquela via quando há mais de um e a que distância está o semáforo. Outra informação que pode ser extraída de um DRS mais complexo é que velocidade o condutor deve manter para avançar o maior número de sinais verdes em sequência numa determinada avenida.

Um DRS também seria muito útil para pedestres que possuem deficiência visual, pois embora vários cruzamentos e travessias para pedestres possuam sinalização adaptada para deficientes visuais, poucas adaptações incluem sinalização sonora. Além disso há muitas travessias que não possuem sinalização semafórica específica para pedestres, deixando assim o deficiente visual dependente de terceiros para saber o momento correto para atravessar a rua.

Ainda outro cenário de aplicação de um DRS é o de uso de veículos autoguiados. Há um grande número de pesquisas nessa área, e embora diversos pesquisadores destaquem o uso de sensores entre os carros autoguiados, ou troca de informação entre os semáforos e os veículos, Diaz-Cabrera, Cerri e Sanchez-Medina (2012) acredita que um cenário híbrido é mais realista, com veículos autoguiados e outros não. Dessa forma ainda haveria a necessidade de uso do semáforo e os veículos autoguiados necessitariam então de um bom DRS para não causar acidentes.

Jensen et al. (2016) reforça que implementar um DRS robusto não é uma tarefa trivial, pois há diversos problemas a serem vencidos para se detectar e reconhecer o estado de um semáforo de forma confiável. Entre os desafios estão a condição do tempo que altera a iluminação do ambiente e dificulta a identificação de qual sinal está ativo no semáforo, semáforos de tipos diferentes (horizontal ou vertical, suspenso ou em poste), o ângulo do qual o semáforo está sendo observado que pode distorcer suas características geométricas, além de outros componentes que se confundem com o semáforo por terem características comuns como a cor.

As limitações apresentadas anteriormente dificulta, e em alguns casos torna impossível, a detecção e reconhecimento de semáforos por métodos convencionais de processamento de imagens. Por exemplo, se o semáforo estiver parcialmente oculto terá a detecção seriamente prejudicada ao utilizar métodos de detecção unicamente baseados em forma. Caso as cores do semáforo estejam alteradas, seja por defeito ou pela iluminação, tornam-se inviáveis os métodos de detecção baseados em cor.

A pesquisa bibliográfica, apresentada na Seção 2.3, mostra que técnicas de Processamento de Imagens - PI e de Aprendizado de Máquina - AM que utilizem os dados obtidos diretamente da imagem tem apresentado resultados de acurácia ou precisão acima de 80%, à exceção dos

momentos em que a limitação ocorre. Sendo assim, uma forma de contornar as limitações pode ser utilizar conhecimento sobre o problema ou sobre o ambiente que não pode ser obtido diretamente da imagem. A este tipo de conhecimento damos neste trabalho a denominação de Conhecimento Especialista, ou Instrução Especialista, uma vez que o conhecimento pode estar implícito na forma que o modelo foi construído e não no dado em si, teoria que dissertação se propõe a validar.

É importante também notar que, havendo uma variedade grande de tipos de semáforos inclusive na mesma cidade, a criação de um DRS que possa ser utilizado em larga escala também é dificultada, pois é inviável coletar e classificar cenas de trânsito para treinamento em cada região. A coleta de dados e categorização destes é um problema recorrente no uso de AM para aplicação tanto em PI como em outras áreas. Isso se dá pois o processo é trabalhoso, muitas vezes pode não haver dados disponíveis, ou não ser possível a coleta em quantidade suficiente e variada. Pensando nisso, o Conhecimento Especialista escolhido também pode ser de ajuda para treinar o algoritmo de AM com uma menor quantidade de dados.

## 1.1 Hipótese

O uso de instrução/conhecimento especialista melhora o processo de detecção e reconhecimento de semáforos. Essa melhora deve ser observada na taxa de precisão do modelo em relação à quantidade de dados necessária para treinar o modelo.

## 1.2 Objetivos

### 1.2.1 Objetivo Geral

Avaliar se o uso de um modelo de detecção e reconhecimento de semáforos que utilize instrução especialista/conhecimento específico obtém resultados melhores que um modelo de detecção e reconhecimento de semáforos sem instrução especialista.

### 1.2.2 Objetivos Específicos

- Analisar um conjunto de dados e extrair instrução especialista desse conjunto
- Criar uma forma de combinar a instrução especialista com um algoritmo de aprendizado de máquina

Para avaliação dos resultados deste trabalho serão utilizadas as métricas acurácia, precisão e cobertura por serem popularmente empregadas neste tipo de pesquisa, conforme constatado na pesquisa bibliográfica realizada. A medida de acurácia  $a$ , de precisão  $p$ , e a medida de cobertura  $c$ , são definidas respectivamente pelas equações 1.1, 1.2 e 1.3, onde  $N_{vp}$  representa o número de



verdadeiros positivos,  $N_{vp}$  o número de verdadeiros negativos,  $N_{fp}$  o número de falsos positivos, e  $N_{fn}$  o número de falsos negativos encontrados pelo modelo.

$$a = \frac{N_{vp} + N_{vn}}{N_{vp} + N_{vn} + N_{fp} + N_{fn}}, \quad 0 \leq a \leq 1, \quad (1.1)$$

$$p = \frac{N_{vp}}{N_{vp} + N_{fp}}, \quad 0 \leq p \leq 1, \quad (1.2)$$

$$c = \frac{N_{vp}}{N_{vp} + N_{fn}}, \quad 0 \leq c \leq 1. \quad (1.3)$$

Esta dissertação está organizada em 5 capítulos. No Capítulo 2 são apresentados conceitos de Processamento de Imagens, Aprendizado de Máquina, e os trabalhos relacionados essenciais para a pesquisa. O Capítulo 3 discute como realizar o Aprendizado de Máquina com IE, sendo a principal contribuição da pesquisa e validando assim a hipótese apresentada. No Capítulo 4 é apresentado o protótipo de um DRS utilizando um smartphone. Por último, as conclusões são realizadas no Capítulo 5.

# 2

## Referencial Teórico

O método de trabalho dessa dissertação inclui a avaliação e experimentação de modelos de AM que possuem como dado a imagem. Considerando isso, este capítulo relaciona conceitos básicos importantes para o bom entendimento do método. Em relação ao Processamento de Imagens são explicadas a formação básica das imagens e alguns passos fundamentais para processá-las. No contexto de Aprendizado de Máquina, foi apresentado o funcionamento dos dois modelos utilizados nos experimentos, a SVM e a CNN que foi utilizada como base para a PCAnet. Também serão apresentados neste capítulo os trabalhos relacionados ao desenvolvido nesta dissertação.

### 2.1 Processamento de Imagens e Visão Computacional

O uso de câmeras para obtenção de imagens importantes em determinado ambiente é comum, e em conjunto com uso da câmera, algoritmos de processamento de imagens podem ser utilizados para o tratamento e obtenção de informações relevantes da cena.

A área de processamento de imagens digitais tem atraído grande interesse nas últimas décadas. A evolução da tecnologia digital, aliada ao desenvolvimento de novos algoritmos, capazes de processar sinais bidimensionais, vem permitindo uma gama de aplicações cada vez maior (MORALES; CENTENO; MORALES, 2003), como por exemplo, na medicina principalmente na ajuda de diagnósticos, cirurgia guiada por computador, em geo-processamento, radares de trânsito, sensoriamento remoto na visualização do clima de uma determinada região, na arquitetura e nas engenharias (elétrica, civil, mecânica), entre outros (MORGAN, 2008). Um exemplo de aplicação bastante interessante e recente é apresentado por Fernandes, Matos e Aragão (2016) que utiliza a extração de características geométricas de uma imagem para dar suporte ao reconhecimento de faces.

Para Gonzalez R. C. e Woods (2010), o interesse em métodos de processamento de

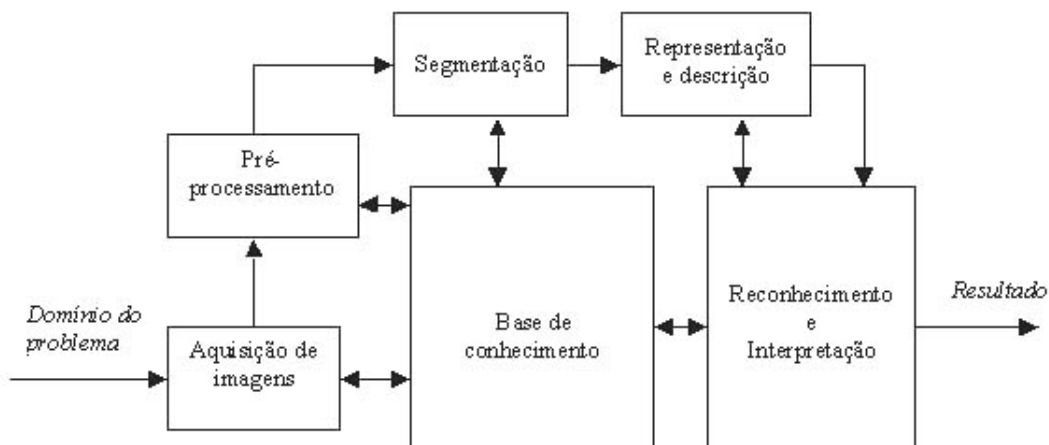
imagens digitais decorre de duas áreas principais de aplicação: melhoria da informação visual para a interpretação humana e processamento de dados para percepção automática através de máquinas. Segundo Grando (2005), a primeira categoria concentra-se em técnicas para melhora de contraste, realce e restauração de imagens danificadas. A segunda categoria concentra-se em procedimentos para extrair de uma imagem informação de forma adequada, para o posterior processamento computacional. Ambas as categorias servem de base na área de Visão Computacional.

Embora às vezes os termos Visão Computacional (VC) e Processamento de Imagens (PI) sejam empregados como sinônimos, a VC pode ser também definida como um caso particular de PI no qual as técnicas empregadas tem o objetivo de entender, descrever, ou dar algum significado a uma imagem ou conjunto de imagens utilizando para isso os valores dos pixels.

### 2.1.1 Passos Fundamentais em Processamento de Imagens

Uma imagem pode ser definida como uma forma compacta de representar muitas informações. Em um sistema de processamento de imagens, essas informações podem passar por diversas etapas, as quais descrevem o fluxo das informações com um dado objetivo definido pela aplicação (GONZALEZ R. C. E WOODS, 2010). A estrutura funcional completa de um sistema de processamento de imagens é ilustrada na Figura 1.

Figura 1 – Estrutura funcional completa de um sistema de processamento de imagens.



Fonte: adaptada de Gonzalez R. C. e Woods (2010)

De acordo com Gevarter (1984), em termos gerais o entendimento de uma imagem em um sistema começa com um *array* de pixels que define a imagem computacionalmente e o uso de modelos que ajudem a determinar o conteúdo da imagem. Tipicamente várias características como linhas e áreas são encontradas. Estas características são então comparadas com os modelos para encontrar uma combinação quando o objetivo é encontrar um objeto específico. Em casos

mais genéricos é necessário determinar várias características da cena e utilizar modelos mais gerais que determinem, por exemplo, formas geométricas, relação entre objetos.

Como afirma Gonzalez R. C. e Woods (2010), o diagrama não significa que todo processo se aplique a uma imagem, pois as metodologias podem ser aplicadas a imagens para diferentes propósitos, com diferentes objetivos. Importante ressaltar que ao aplicar todas as fases do diagrama, em especial as três últimas, obtemos ao mesmo tempo um sistema de visão computacional. Sendo assim, segue uma descrição das etapas mais comuns. As descrições destas etapas foram norteadas principalmente por Facon (2002) e Gonzalez R. C. e Woods (2010), sendo as seguintes:

- **Aquisição da imagem:** consiste em adquirir uma imagem através de um sensor e transformá-la em uma imagem digital, sob a forma de uma tabela de valores discretos inteiros chamados de *pixel*. Dentre os aspectos envolvidos neste passo pode-se mencionar: a escolha do tipo do sensor, o conjunto de lentes a utilizar, as condições de iluminação da cena, os requisitos de velocidade da aquisição, a resolução e o número de níveis de cinza da imagem digitalizada, entre outros;
- **Pré-processamento:** a imagem resultante do passo anterior pode apresentar diversas imperfeições, tais como presença de pixels ruidosos, contraste e/ou brilho inadequado, regiões interrompidas ou indevidamente conectadas por sombras ou sobreposição, entre outras. Assim, a função do pré-processamento é melhorar a imagem de forma a aumentar as chances para o sucesso dos processos seguintes. Este passo envolve técnicas para filtragem e realce, remoção de ruído, compressão, e etc. O pré-processamento não é indispensável, mas necessário na maioria dos casos;
- **Segmentação:** consiste em dividir uma imagem em partes ou objetos constituintes, ou seja, nos objetos de interesse que compõem a imagem. A segmentação é efetuada pela detecção de descontinuidades (contornos) e/ou de similaridades (regiões) na imagem. Em geral, a segmentação automática é uma das tarefas mais difíceis no processamento de imagens digitais. Por um lado, um procedimento de segmentação de imagens bem-sucedido aumenta as chances de sucesso na solução de problemas que requerem que os objetos sejam individualmente identificados. Por outro lado, algoritmos de segmentação inconsistentes quase sempre acarretam falha no processamento;
- **Representação e descrição:** o alvo da representação é elaborar uma estrutura adequada, agrupando os resultados das etapas precedentes. A representação pode ser por fronteira e/ou regiões. A representação por fronteira é adequada quando o interesse se concentra nas características externas (cantos ou pontos de inflexão). A representação por região é adequada quando o interesse se concentra nas propriedades internas (textura ou forma do esqueleto). O processo de descrição, também chamado de seleção de características,

procura extrair características que resultam em informação quantitativa ou que sejam básicas para a discriminação entre classes de objetos;

- **Reconhecimento e interpretação:** reconhecimento é o processo que atribui um rótulo a um objeto, baseado na informação fornecida pelo descritor. A interpretação envolve a atribuição de significado a um conjunto de objetos reconhecidos. É o passo mais elaborado do processamento de imagens digitais, pois permite obter a compreensão e a descrição final do domínio do problema, fazendo uso do conhecimento a priori e do conhecimento adquirido durante as fases precedentes;
- **Base de conhecimento:** o processamento de imagens digitais pressupõe a existência de conhecimento prévio sobre o domínio do problema, armazenado em uma base de conhecimento, cujo tamanho e complexidade variam dependendo da informação. Embora nem sempre presente, a base de conhecimento guia a operação de cada módulo do processamento, controlando a interação entre os módulos.

É possível perceber, à medida que se passa por níveis crescentes de abstração, que ocorre uma redução progressiva da quantidade de informações manipuladas. Na aquisição da imagem e no pré-processamento, os dados de entrada são *pixels* da imagem original e os dados de saída representam propriedades da imagem na forma de valores numéricos associados a cada pixel. Na segmentação, representação e descrição, esse conjunto de valores produz como resultado uma lista de características. O reconhecimento e a interpretação produzem, a partir dessas características, uma interpretação do conteúdo da imagem (FACON, 2002). Uma variedade de paradigmas, ou métodos, tem sido propostos para realizar as tarefas descritas anteriormente, sendo os principais os paradigmas *Top-Down* e *Bottom-up* (GEVARTER, 1984).

Pereira (2007) e Gevarter (1984) definem de forma didática os dois métodos. O método *top-down* usa conhecimentos obtidos a priori para detectar regiões de maior interesse numa imagem. Esses conhecimentos podem ser obtidos de várias formas. Geralmente, utilizam-se ferramentas de aprendizagem baseadas em modelos geométricos/relacionais (como redes semânticas ou grafos relacionais) ou modelos estatísticos (como redes neurais e máquinas de vetores de suporte). Porém, esses conhecimentos também podem ser fornecidos por um ser humano, selecionando-se manualmente regiões de maior interesse numa imagem. Um exemplo de aplicação desse método é o uso de casamento de templates (*template matching*). O paradigma *bottom-up* é guiado por características primitivas da imagem como cor, intensidade e orientação. Além disso, ele atua de modo inconsciente, ou seja, o observador é levado a fixar sua atenção em determinadas regiões da imagem devido aos estímulos causados pelos contrastes entre características visuais presentes na imagem.

### 2.1.2 Fundamentos de Imagens Digitais

Para Jayaraman (2011), uma imagem pode ser definida como uma função bidimensional que representa a medida de características tais como brilho ou cor de uma determinada cena. Uma imagem é uma projeção de uma cena 3D em um plano 2D. Uma imagem é definida matematicamente como uma função de duas variáveis  $f(x,y)$  que define a intensidade da luz no ponto  $(x,y)$ . O valor da intensidade da luz, ou brilho, definido por  $f(x,y)$ , é conhecido como *pixel*. É importante notar que o processamento de uma imagem depende diretamente destes valores. O termo *pixel* é uma abreviatura do inglês *picture element* que significa elemento da figura, corresponde a menor unidade de uma imagem digital, onde são descritos a cor e o brilho específico de uma célula da imagem (MORGAN, 2008).

Normalmente, uma imagem capturada por uma câmera de vídeo é apresentada em cores, assim, cada pixel da imagem deve ser formado por um conjunto de valores, ou também conhecido como canais de cores, geralmente de tamanho três ou quatro, podendo pertencer aos padrões de cores *RGB* (*red*, *green* e *blue*) ou *CMYK* (*cyan*, *magenta*, *yellow* e *key=black*), respectivamente. Cada canal de cor, em geral, possui uma variação que vai de 0 a 255. Há também o canal de cor *HSV* ou *HSB*, sigla para *hue* (matiz), *saturation* (saturação), e *value* (valor) ou *brightness* (brilho), respectivamente. Neste caso, o matiz representa a cor em si e assume valor entre 0 e 360, a saturação define a pureza da cor e, quanto menor o valor, mais próximo de cinza a cor será, e o valor define o brilho da cor. A saturação e o valor assumem valores entre 0 e 1.

A combinação dos canais de cores pode representar uma grande variedade de cores, no entanto, muitas vezes essa quantidade de informação pode ser desnecessária para os objetivos de determinadas aplicações, de modo que seu processamento possa levar a desperdício de recurso. Uma forma para a resolução deste problema é a utilização de uma técnica de transformação para tons de cinza, considerado um processo simples sob os canais de cores. A Equação 2.1, apresentada a seguir, descreve a transformação dos canais de um pixel  $i$ , pertencente ao padrão *RGB*, para tons de cinza, como segue:

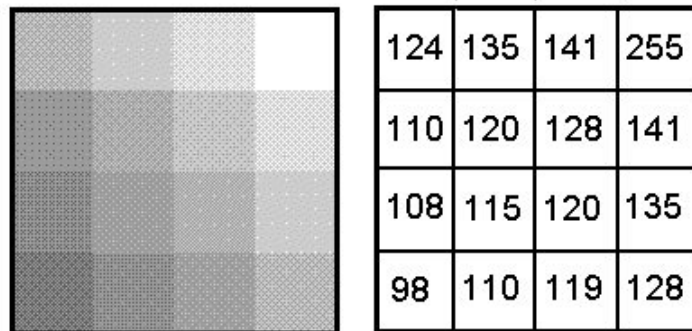
$$I_i = \frac{R_i + G_i + B_i}{3}, \quad (2.1)$$

em que é calculada a média aritmética dos três canais de cores,  $R$ ,  $G$ , e  $B$  do *pixel*  $i$ .  $I_i$  é o valor do tom de cinza obtido, também conhecido como valor de intensidade, representando o *pixel*  $i$  por um único valor.

Como afirma Grando (2005) uma imagem digital  $f(x,y)$  pode ser representada por uma matriz, cujos índices de linha e coluna identificam um ponto (*pixels*) da imagem e representam o conjunto de valores (canais de cores). Por exemplo, a Figura 2 representa uma imagem digital de 4 pixels de largura por 4 pixels de altura, cujos elementos são dados pelas intensidades dos pixels nas posições correspondentes.



Figura 2 – Imagem digital e respectiva representação matricial.



Fonte: Almeida, A. B. (1998).

## 2.2 Aprendizado de Máquina

Uma das fases mais importantes de um sistema de processamento de imagens completo, ou sistema de visão computacional, é a fase de reconhecimento, muitas vezes aliada à uma base de conhecimento como pode-se visualizar na Figura 1. Nesta seção discutiremos sobre conceitos importantes de Aprendizado de Máquina (AM), uma vez que é uma ferramenta bastante usada para reconhecimento em vários tipos de sistema incluindo a área de VC.

O aprendizado é um processo que inclui a aquisição de conhecimento, o desenvolvimento de habilidades motoras e cognitivas através de instrução ou prática, organização de novo conhecimento em conhecimento generalizado, representação do conhecimento, e descobrimento de novos fatos e teorias através de observação e experimentação (MICHALSKI; CARBONELL; MITCHELL, 2013).

A utilização de AM pode ser entendida como uma forma de simular o aprendizado humano em uma máquina computacional aplicando para isso as atividades ligadas ao aprendizado como memorização, observação e exploração.

De acordo com Carvalho e Faceli (2011) a programação de computadores de forma que eles aprendam com a experiência passada é baseada num princípio de inferência denominado indução. Indução é o processo no qual conclusões genéricas são geradas a partir de uma série de exemplos. Segundo Goodfellow, Bengio e Courville (2016), um algoritmo de aprendizado de máquina é capaz de aprender a partir de dados. Algoritmos de AM aprendem então a induzir funções ou hipóteses capazes de resolver problemas a partir de dados que representam instâncias do problema a ser resolvido. Esses dados formam o conjunto de dados, ou base de conhecimentos, que aparece na Figura 1.

Uma lista de aplicações bem-sucedidas de técnicas de AM na solução de problemas reais pode ser vista a seguir ((CARVALHO; FACELI, 2011)):

- Reconhecimento de fala;

- Predição de taxas de cura de pacientes com diferentes doenças;
- Detecção do uso fraudulento de cartões de crédito;
- Condução de automóveis de forma autônoma em rodovias;
- Ferramentas que jogam gamão e xadrez de forma semelhante a campeões;
- Diagnóstico de câncer por meio da análise de dados de expressão gênica;
- Detecção de placas no trânsito;
- Análise de sentimentos em textos.

Como será visto na Seção 2.3, AM é uma das principais técnicas utilizadas para alguns tipos de reconhecimento de objetos em imagens, senão em todos, e alguns dos algoritmos mais empregados para esse fim são Redes Neurais e *Support Vector Machine (SVM)* também conhecidas como Máquinas de Vetores de Suporte, em tradução livre.

### 2.2.1 Redes Neurais Artificiais

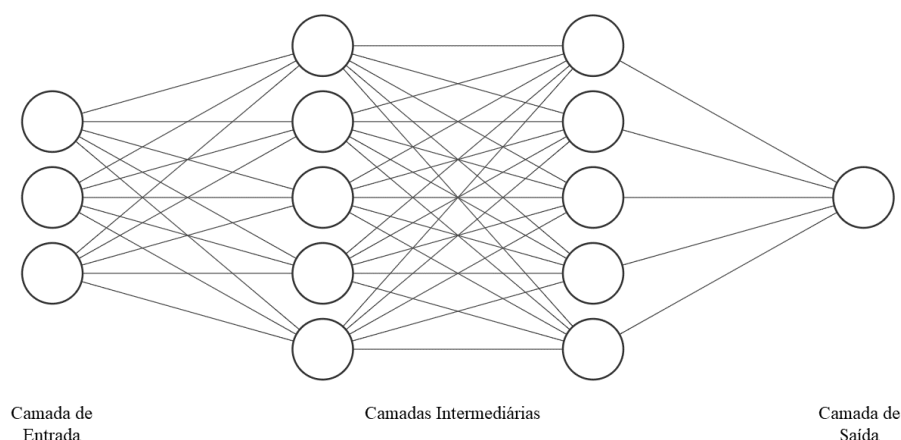
Na busca pela construção de máquinas inteligentes, ou com comportamento inteligente, exemplo que surge naturalmente é o do cérebro humano. O sistema nervoso humano contém em torno de 8 bilhões de neurônios, sua célula fundamental, e eles estão conectados por aproximadamente  $10^{14}$  ligações ou sinapses (Karpathy, A., 2017). Cada um desses neurônios processa e se comunica com milhares de outros continuamente e em paralelo. A estrutura desses neurônios, a topologia das suas conexões e o seu comportamento em conjunto formam a base para o estudo das Redes Neurais Artificiais (RNA ou RN) (BRAGA, 2007).

Algumas similaridades importantes entre o sistema nervoso biológico e uma RNA apontadas por Braga (2007) são o fato de que os dois sistemas são baseados em unidades de computação paralela e distribuída que se comunicam entre si, e ambos possuem detetores de características, redundância e modularização das conexões.

Segundo Carvalho e Faceli (2011) as RNAs podem ser definidas como sistemas computacionais distribuídos compostos de unidades de processamento simples, densamente interconectadas. As unidades são conhecidas como neurônios, assim como no sistema nervoso biológico, e computam funções matemáticas. Os neurônios são então dispostos em uma ou mais camadas e interligados por um grande número de conexões. Na Figura 3 pode-se visualizar a representação de uma RNA de múltiplas camadas com 1 camada de entrada, 2 camadas intermediárias, e 1 camada de saída.

Em grande parte das arquiteturas de RNAs essas conexões possuem pesos associados, que ponderam a entrada recebida por cada neurônio da rede. Os pesos tem seus valores ajustados

Figura 3 – Exemplo de RNA multicamadas típico.



Fonte: Elaborado pelo autor.

em um processo de aprendizado e codificam o conhecimento adquirido pela rede (BRAGA, 2007).

Uma arquitetura de RNA bastante interessante para pesquisa em VC é a *Convolutional Neural Network* (CNN). As CNNs assumem explicitamente que as entradas serão imagens, o que permite que determinadas propriedades relacionadas a isso possam ser incluídas na arquitetura, o que reduz a quantidade de parâmetros que a rede precisa (Karpathy, A., 2017).

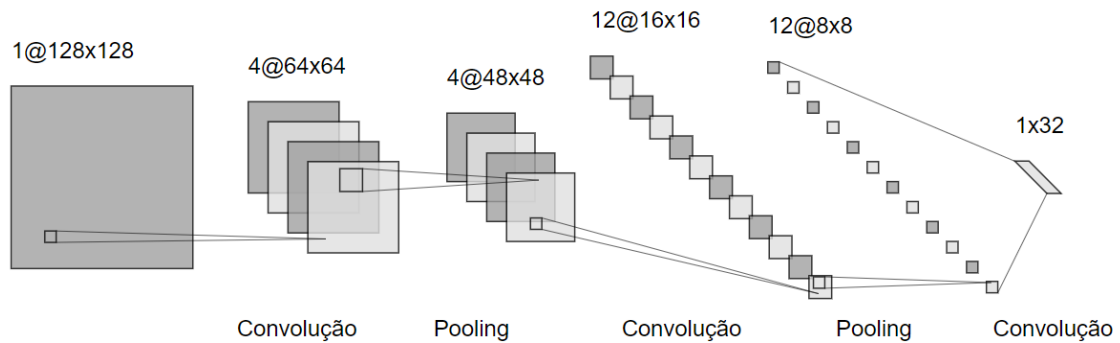
De acordo com Haykin et al. (2009), uma CNN típica contém dois tipos principais de camadas: camada de convolução e camada de amostragem ou de *pooling*. Na camada de convolução, um kernel de tamanho fixo é utilizado para calcular a média dos produtos dos pixels de uma parte da imagem de tamanho igual ao kernel, esse processo é realizado em toda a imagem de entrada da camada.

A camada de convolução é quase sempre seguida de uma camada de pooling. Segundo Goodfellow, Bengio e Courville (2016), a camada de pooling é responsável por substituir um pixel de uma determinada saída em uma das camadas da rede por um resumo estatístico da sua vizinhança. O objetivo dessa tarefa é tornar a rede menos sensível a pequenas diferenças na imagem de entrada, como por exemplo uma leve rotação na imagem. Na Figura 4 é possível observar a arquitetura de uma CNN de 6 camadas, sendo 1 camada de entrada, 4 camadas intermediárias e 1 camada de saída.

Um excelente exemplo do uso de CNN ao trabalhar com imagens pode ser observado no trabalho de Affonso et al. (2017) onde a rede neural é aplicada para classificar a qualidade de tábuas de madeira baseado em suas imagens.

Embora bem adaptada para trabalhar com imagens, a CNN também pode trabalhar com

Figura 4 – Exemplo de arquitetura de uma CNN.



Fonte: Elaborado pelo autor.

outros tipos de dados que possuam uma topologia conhecida, semelhante a uma grade, ou matriz (GOODFELLOW; BENGIO; COURVILLE, 2016). Por exemplo, no trabalho de Júnior et al. (2016) uma CNN foi utilizada para a tarefa de reconhecimento de entidades nomeadas em textos em língua portuguesa, enquanto Santos et al. (2015) empregou a mesma arquitetura para o problema de reconhecimento de fala em ambientes ruidosos.

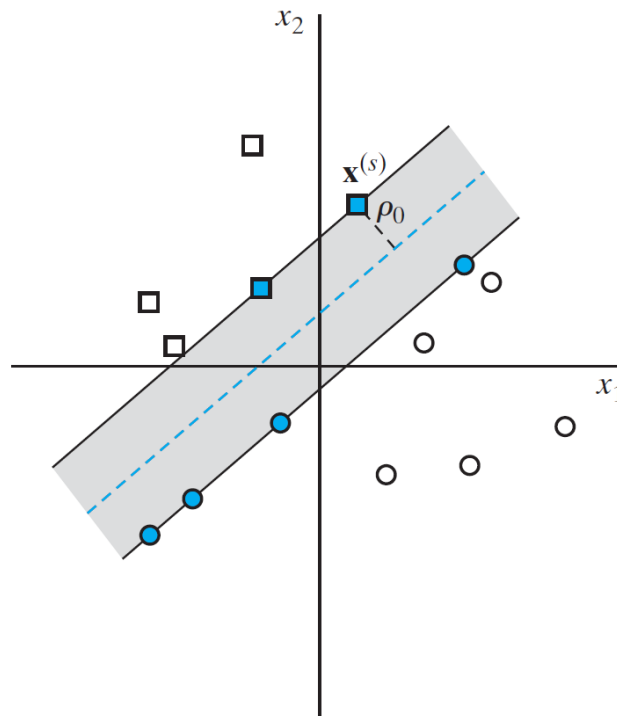
### 2.2.2 Support Vector Machines

Support Vector Machines (SVM), ou Máquina de Vetores de Suporte, é uma técnica de AM baseada em Teoria do Aprendizado Estatístico (TAE). A TAE estabelece condições matemáticas que auxiliam na escolha de um classificador particular  $\hat{h}$  a partir de um conjunto de dados de treinamento. Essas condições levam em conta o desempenho do classificador no conjunto de treinamento e sua complexidade, com o objetivo de obter um bom desempenho também para novos dados do mesmo domínio (CARVALHO; FACELI, 2011).

O princípio básico da SVM, de acordo com Haykin et al. (2009), é a possibilidade de dividir um conjunto de dados contendo duas classes utilizando um conjunto de pontos no espaço, formando assim um vetor ou hiperplano de separação entre esses dados. O objetivo da SVM é encontrar esse hiperplano e maximizar a margem de separação, que consiste na distância entre o hiperplano que separa os dados e o dado mais próximo de cada um dos seus lados, ou seja, de cada uma das classes. Maximizando-se essa distância encontra-se o que é chamado de hiperplano ótimo. Na Figura 5 observa-se a representação de um hiperplano ótimo para dados linearmente separáveis em duas classes,  $x^{(s)}$  representa os dados de entrada,  $\rho_0$  representa a margem de separação, e a linha pontilhada representa o hiperplano ótimo encontrado. O treinamento de uma SVM é supervisionado, ou seja, é necessário o uso de dados previamente classificados.

Segundo Shi, Zou e Zhang (2016) a SVM é uma técnica de popularidade crescente por

Figura 5 – Ilustração de um hiperplano ótimo encontrado por uma SVM.



Fonte: Haykin et al. (2009).

possuir boa capacidade de generalização além de conseguir apresentar resultados comparáveis e muitas vezes superiores aos de outros algoritmos de AM, como as RNAs.

As SVMs também podem ser utilizadas para problemas de classificação com mais de duas classes, embora nesses casos seja grande a probabilidade dos dados não serem linearmente separáveis. Luger (2005) mostra que uma das formas de fazer isso é por treinar repetidamente a SVM em cada classe de interesse contra todas as outras classes.

Exemplos de aplicação de SVMs podem ser visualizados nos trabalhos de Lorena e Carvalho (2005) com o objetivo de identificar a localização da proteína no interior de uma célula, de Fernandes, Matos e Aragão (2016) para reconhecimento de faces, e de Affonso et al. (2017) para classificação da qualidade de tábuas de madeira.

## 2.3 Revisão Bibliográfica

Um mecanismo reconhecedor de objetos numa imagem em geral é formado por duas partes: uma fase inicial de detecção que identifica possíveis objetos, e uma fase posterior de classificação desses possíveis objetos, classificando-os como pertencente ou não à determinada classe de objeto. Uma terceira fase não obrigatória é a fase de rastreamento de objetos que tem se tornado popular nos últimos anos para o problema de reconhecimento de semáforos.

Observando-se trabalhos disponíveis em bibliotecas digitais podemos citar algumas das

técnicas de detecção e reconhecimento de objetos quando o objeto em questão é um semáforo. Foram pesquisados trabalhos nos sites do Portal de Periódicos da CAPES, do Google Acadêmico, da *ACM Digital Library*, e da *IEEE Xplore Digital Library*. É importante ressaltar que quando se trabalha com detecção e reconhecimento de objetos é necessário definir características do objeto que serão indispensáveis para o seu reconhecimento, independente de qual técnica seja utilizada.

No caso do semáforo essas características são cores, forma e luz. Embora a dependência entre esses componentes possa ser considerada uma característica intrínseca ao objeto não a incluiremos nessa categoria por se tratar de uma informação mais complexa obtida relacionando as outras características do semáforo.

As técnicas mais utilizadas para detecção de semáforo são Redes Neurais, Mapa de Saliência, Transformadas, e Detecção de *Blob*.

Weber, Wolf e Zöllner (2016), John, Zheming e Mita (2016), John et al. (2015), John et al. (2014) utilizaram Redes Neurais Convolucionais (CNN - Convolutional Neural Network) para detectar o semáforo, enquanto Wang et al. (2015) utilizaram uma Rede Neural PCAnet com o mesmo objetivo.

Philipsen et al. (2015) utilizaram um algoritmo de aprendizado baseado nos canais de características da imagem e nos histogramas de gradientes orientados para detecção e reconhecimento.

Mapas de Saliência são utilizados como ferramenta de detecção por Kim et al. (2016), Almeida et al. (2016), Almeida, Vasconcelos e Benicasa (2015), Ji et al. (2015) e John et al. (2015). Já Zhou, Chen e Huang (2016), Chen e Huang (2016), Zhang et al. (2014) empregam um algoritmo de Detecção de *Blob*.

Algumas das transformadas geométricas utilizadas na fase de detecção são a Transformada Circular de Hough usada por Widyanoro e Saputra (2015), Chiu, Chen e Hsieh (2014), e Omachi e Omachi (2010); e a Transformada Rápida de Simetria Radial por Sooksatra e Kondo (2014).

Outras técnicas menos comuns também são utilizadas sozinhas ou associadas com as citadas anteriormente, como Filtros Adaptativos por Shi, Zou e Zhang (2016), *Template Matching* por Trehard et al. (2014), Distribuição Gaussiana por Oniga, Prodan e Nedevschi (2015), Estimativa de probabilidade de classe associada à CNN por Weber, Wolf e Zöllner (2016), e *Top Hat* por Jie et al. (2013).

O uso de algoritmos de processamento de imagens como uma das técnicas principais também é comum. Observa-se o uso de segmentação de cor ou de forma nos trabalhos de Borrmann et al. (2014) e Zong e Chen (2014), o uso associado com limiarização é feito por Sathiya, Balasubramanian e Priya (2014) e por Gong et al. (2010).

Na fase de reconhecimento a maioria dos trabalhos emprega algoritmos de aprendizado



de máquina como Redes Neurais, *Support Vector Machines* - SVM, e Lógica Fuzzy.

Chen e Huang (2016) utiliza CNN, já Weber, Wolf e Zöllner (2016) utiliza uma PCA-network, um tipo de rede neural que simula o funcionamento de uma CNN com menos camadas.

Shi, Zou e Zhang (2016), Zhou, Chen e Huang (2016), Shi, Zhao e Xia (2016), Michael e Schlippsing (2015), Chen e Huang (2016), Wang et al. (2015), Ji et al. (2015), Salarian, Manavella e Ansari (2015), Jang et al. (2014), Chen, Shi e Zou (2014) empregam SVMs para reconhecer o semáforo, em alguns casos aliadas à uma rede neural. Ao mesmo tempo o uso de *Fuzzy* para classificação pode ser observado em Almeida et al. (2016) e Chung, Wang e Chen (2002)

Algumas outras técnicas são utilizadas nessa fase como substitutas ao aprendizado de máquina, como um auxílio final eliminando falsos positivos, ou fazendo a ligação entre o resultado da fase de detecção e a entrada da fase reconhecimento. Zhou, Chen e Huang (2016), Michael e Schlippsing (2015), Ji et al. (2015), Almeida, Vasconcelos e Benicasa (2015), e Almagambetov, Velipasalar e Baitassova (2015) usaram Histogramas, este último tendo como objeto o semáforo para pedestres especificamente. Balcersek et al. (2014), Cai, Li e Gu (2012), Omachi e Omachi (2010) e Omachi e Omachi (2009) usaram Transformadas. John et al. (2015), Choi, Ahn e Kweon (2013), Fan, Lin e Yang (2012), e Charette e Nashashibi (2009) usaram *Template Matching*. John et al. (2014) usou Mapa de Saliência, e Levinson et al. (2011) usou Histogramas com Probabilidade.

O uso de Correlação Cruzada Normalizada pode ser observada no trabalho de Mascetti et al. (2016) aplicada ao reconhecimento de semáforos para pedestres. *Hidden Markov Models* - HMM são aplicados para reconhecimento de semáforos comuns por Gomez et al. (2014).

Observando características como disponibilidade dos dados para reprodução, ano da publicação, e relevância do canal de publicação foram selecionados alguns artigos para possível reprodução e/ou aproveitamento de técnicas no presente trabalho. A seguir os trabalhos selecionados são abordados em detalhes.

Chen e Huang (2016) propõem um novo método que combina técnicas de visão computacional e de aprendizado de máquina para reconhecer semáforos. Extração de cor e detecção de *blob* são utilizados para detectar possíveis semáforos. A extração de cor é realizado no espaço de cor HSV. A detecção de blob é implementada combinando algoritmos de processamento de imagens como inundamento, seguimento de contorno, e fechamento. A combinação destas técnicas permite que tanto semáforos circulares quanto semáforos com setas possam ser identificados.

Após a detecção das regiões de interesse um classificador PCA é aplicado. O classificador PCA proposto é formado por uma rede PCA e uma SVM linear. Uma rede PCA (PCAnet) é uma rede neural profunda que simula o comportamento de uma CNN tradicional.

A PCAnet é baseada na técnica estatística conhecida como Análise de Componente Principal (*Principal Component Analysis* - PCA). De acordo com Hair et al. (2006), um PCA é uma abordagem estatística que pode ser utilizada para analisar relações entre um grande número

de variáveis e encontrar uma forma de reduzir a informação a partir de um conjunto original de variáveis para um conjunto menor com perda mínima de informação. Isso permite que a PCAnet simule o comportamento de uma Rede Neural Convolucional (*Convolutional Neural Network - CNN*) com menos custo computacional.

De forma geral a convolução na PCAnet é realizada utilizando os filtros PCA como *kernel*, estes são encontrados por aprendizado não-supervisionado durante o processo de treinamento. O número de filtros PCA pode ser variante e, de acordo com Chen e Huang (2016), quanto mais filtros PCA melhor a performance da rede. No trabalho em questão foram utilizados 8 filtros PCA nos suas duas camadas.

Para cada conjunto de pixels na imagem de entrada, há um conjunto de pixels de mesmo tamanho no filtro. A média deles é calculada e o valor é removido do filtro, se houver. Em seguida é realizada a convolução com os filtros PCA. A combinação de cálculo de média e convolução são encarados como estágios, ou camadas, da rede. A saída da PCAnet contém valores positivos e negativos, que são binarizados e em seguida convertidos para valores decimais. Um vetor de características é formado com esses valores e são utilizados então como entrada da SVM.

São criadas 5 redes PCA (quantidade de grupos de semáforos identificados) para estimar pesos ou *features* capazes de identificar os tipos de semáforo. Os pesos então alimentam uma SVM que faz a classificação final. Os testes são realizados utilizando um *dataset* produzido e disponibilizado publicamente pelos autores. Esse e outros *datasets* são apresentados na subseção seguinte.

Shi, Zou e Zhang (2016) propõem um algoritmo adaptativo de supressão de *background* com o objetivo de realçar regiões de interesse, o AdaBSF. No algoritmo proposto inicialmente são calculados os gradientes normalizados da imagem de entrada. Combinando os gradientes normalizados com os valores das camadas R, G, e B da imagem original obtém-se um mapa de características simples de 4 camadas. Cada janela de detecção  $W_i$  neste mapa de características pode ser representado como um vetor de características básico  $x_i \in \mathbb{R}^{D \times 1}$ . Na busca por semáforos verticais e horizontais o tamanho da janela é fixado em  $16 \times 8$  pixels e  $8 \times 16$  pixels, respectivamente. Considerando isso, a dimensão de cada janela de detecção é  $D = 16 \times 8 \times 4 = 512$ . Para resolver o problema de multi-escala no qual o semáforo pode aparecer em diferentes tamanhos na imagem, a imagem original é redimensionada para diferentes escalas enquanto a janela de detecção é mantida do mesmo tamanho.

O objetivo do algoritmo AdaBSF é especificar um filtro de Resposta de Impulso Finito especificado pelo vetor  $w = [w_1, w_2, \dots, w_{256}]^T$  de forma que  $y_i = w^T x_i$ . O valor  $y_i$  representa um *score* para cada janela de detecção que representa a probabilidade de aquela janela ser parte de um semáforo,  $x_i$  representa os pesos aprendidos pelo algoritmo (SHI; ZOU; ZHANG, 2016).

Na fase de reconhecimento cada região candidata é verificada e classificada em diferentes classes de semáforo utilizando uma SVM. O histograma local de cor e o histograma orientado a

gradiente de cada região candidata são utilizados como descritores de características do SVM linear e são usados para treinar o modelo e para classificar as regiões encontradas pelo AdaBSF. Os testes são realizados com um dataset próprio, disponibilizado pelo autor e discutido na seção seguinte.

### 2.3.1 Datasets

Um dos maiores desafios ao se trabalhar com reconhecedores de padrões, seja em imagens ou outros campos, é o de montar ou encontrar bases de dados adequadas para a pesquisa. Apesar de ser uma área de pesquisa relativamente nova com o primeiro trabalho encontrado datando de 2002, foram encontrados alguns bancos de dados interessantes.

Charette e Nashashibi (2009) utilizam um dataset que tornou-se referência na área de reconhecimento de semáforo para veículos (TLR - do inglês *Traffic Light Recognition*). Conhecido como LaRA - *La Route Automatisée*, o dataset é mantido por *Robotics Centre of Mines ParisTech* e está disponível publicamente em <http://www.lara.prd.fr/benchmarks/trafficlightsrecognition>. O LaRA possui 11179 *frames* de resolução 640x480, e 9168 semáforos categorizados manualmente, sendo 3381 “verdes”, 58 “laranjas”, 5280 “vermelhos”, e 449 “ambíguos”, ou seja quando um ser humano não conseguiu classificá-lo pois houve dúvida.

Outro dataset bastante completo encontrado foi o *WPI Traffic Light Dataset*, mantido pelo *Worcester Polytechnic Institute Embedded Computing Laboratory* e disponibilizado publicamente em <http://computing.wpi.edu/dataset.html>. O WPI possui dois datasets de treinamento, um de 77.8 MB e um opcional de 12 GB, além de um dataset de teste de 3.06 GB, com *frames* de resolução 1920x1080 categorizados manualmente. O WPI foi avaliado por Chen e Huang (2016) e teve os resultados comparados com os obtidos por Charette e Nashashibi (2009) na mesma ocasião. O *Worcester Polytechnic Institute Embedded Computing Laboratory* também disponibiliza no mesmo link o *WPI Pedestrian Dataset*, bastante útil para testes do TLR buscando um semáforo levemente alterado como no caso do semáforo para pedestres.

Shi, Zou e Zhang (2016) apresentam um outro dataset mantido pelo *LEVIR -LEarning, Vision and Remote sensing laboratory* e disponibilizado publicamente em <http://levir.buaa.edu.cn/Code.htm>. O LEVIR dataset contém 5654 *frames* categorizados para teste, divididos em 8 sequencias de imagens. Foram categorizados 2713 semáforos verdes e 5245 semáforos vermelhos nessas 8 sequências. Shi, Zou e Zhang (2016) também disponibiliza 10975 imagens de semáforos verdes e 9997 imagens de semáforos vermelhos para treinamento. Os resultados obtidos utilizando esse dataset também já foram comparados por Shi, Zou e Zhang (2016) com os resultados obtidos por Charette e Nashashibi (2009).

# 3

## Aprendizado de Máquina com Instrução Especialista

Algoritmos de Inteligência Artificial são comumente usados para resolver problemas de classificação de objetos, seja em imagem ou em outros campos de pesquisa. Mais especificamente, técnicas de Aprendizado de Máquina tem sido utilizadas em larga escala para generalizar dados, encontrar padrões e classificar objetos, conforme demonstrado na Seção 2.3 sobre a área de reconhecimento de semáforos.

Segundo Haykin et al. (2009) o principal problema que algoritmos de aprendizado de máquina enfrentam é a quantidade de dados necessária para aprender de forma correta. Além da necessidade de uma quantidade razoavelmente grande de dados, estes precisam ser bem balanceados e categorizados. A construção de um conjunto de dados com os requisitos citados requer tempo e esforço, e pode ser inviável considerando a raridade do dado em algum caso específico.

Uma vez que o conjunto de dados está pronto, existe o problema de analisá-lo. Como citado anteriormente, algoritmos de aprendizado precisam de uma grande quantidade de dados para encontrar padrões, ou seja, as regras que generalizam os dados e permitirão que o algoritmo classifique corretamente novos dados. Por isso, o aprendizado leva tempo, o qual pode ser reduzido com o uso de computadores com alto poder de processamento. Importante ressaltar que computadores com alto poder de processamento costumam ser caros e, por esse motivo, não ser acessíveis a uma grande parcela da população.

Uma forma de reduzir a quantidade de dados necessários para treinar o algoritmo é introduzir no processo de treinamento alguma informação que se saiba de antemão sobre o conjunto de dados ou sobre o problema. Haykin et al. (2009) chama este tipo de informação de *prior knowledge*. Esse mesmo conhecimento é denominado Instrução Especialista (IE) neste trabalho.

O tipo de IE que naturalmente vêm à mente quando se pensa em qualquer componente

relacionado à trânsito é a geolocalização, também conhecida como *Global Positioning System* (GPS). O GPS como IE pode ser usado de diferentes maneiras, associado a diferentes técnicas. Alguns trabalhos consideram a exata posição do semáforo e a comparam com a posição do veículo, neste caso pode-se considerar o conhecimento prévio de todas as posições do semáforo como Yoneda, Suganuma e Aldibaja (2016), ou utilizar um algoritmo de mapeamento de semáforos como Levinson et al. (2011), Li-tian et al. (2014) e Fairfield e Urmson (2011).

Outra forma de utilizar o GPS é verificar se o veículo está posicionado próximo a uma interseção onde a chance de existir um semáforo é maior, este conhecimento pode ser explorado para, por exemplo, ativar o mecanismo de reconhecimento de semáforos apenas próximo a interseções como observado por Zhang et al. (2014), o que diminui consideravelmente a quantidade de processamento necessário. John et al. (2014) também fazem uso desse tipo de informação para ajudar na fase de reconhecimento, como um eliminador de falsos positivos ao fim do processamento.

A relação entre os objetos geométricos que formam o semáforo também pode ser considerada como conhecimento especialista. É de conhecimento comum que um semáforo básico é formado por círculos, na maioria das vezes 3, posicionados horizontalmente ou verticalmente. Normalmente estes círculos estão posicionados no interior de um retângulo escuro, também posicionado horizontalmente ou verticalmente. Ying, Tian e Lei (2015) e Charette e Nashashibi (2009) fizeram uso de algumas dessas características.

Sung e Tsai (2013) combinam esses dois tipos de conhecimento especialista num sistema que requer pouco poder de processamento e pode ser executado num smartphone por exemplo.

Considerando o problema apresentado, neste Capítulo será apresentada uma abordagem que valida a hipótese levantada nesta dissertação de que é possível reduzir a quantidade de dados necessários para treinar um algoritmo de aprendizado de máquina utilizando para isso regras definidas por um especialista na área do problema. No contexto dessa pesquisa essas regras serão referenciadas como Instrução Especialista ou Conhecimento Especialista.

Foi necessário implementar um modelo de detecção e reconhecimento de semáforos que utilize conhecimento especialista e compará-lo com os resultados obtidos por outros autores que não utilizaram conhecimento especialista. A comparação foi realizada utilizando *datasets* públicos e utilizados por Chen e Huang (2016) e Shi, Zou e Zhang (2016) em seus trabalhos.

Os resultados obtidos pelo Experimento 1 deste Capítulo usando o *dataset* de Shi, Zou e Zhang (2016), apresentado na Seção 3.1.1, foram publicados no periódico *Information* na edição especial *Information Technology: New Generations (ITNG 2018)*. O artigo publicado tem o título *Prototyping a Traffic Light Recognition Device with Expert Knowledge* e pode ser acessado de forma gratuita pelo link <https://www.mdpi.com/2078-2489/9/11/278>.

## 3.1 Metodologia de Avaliação

Esta seção apresentará a metodologia utilizada para realizar os experimentos descritos nas seções seguintes. Todos os experimentos foram executados num notebook Dell Inspiron 15 7000, processador i7 7ª geração e 16 Gigabytes de memória RAM.

### 3.1.1 Experimento 1

Neste experimento foram utilizados os dados de dois *datasets*, o primeiro disponibilizado por Shi, Zou e Zhang (2016), e o segundo disponibilizado por Chen e Huang (2016). Cada passo realizado na elaboração deste experimento foi executado individualmente em cada *dataset*.

Inicialmente foi necessário elaborar a instrução especialista. Para tanto, os dados de teste cada dataset disponível foram divididos em 2 grupos, um grupo para teste e outro grupo para ser utilizado para gerar a instrução especialista. Deste segundo grupo foram selecionadas aleatoriamente 650 imagens para serem utilizadas na elaboração do conhecimento especialista. Sendo 2 o número de datasets disponíveis, foram analisadas 1300 imagens e gerados 2 conjuntos de instrução especialista.

O algoritmo de classificação de Chen e Huang (2016) foi implementado e utilizado para comparar a influência da instrução especialista nos resultados do treinamento. A explicação detalhada do algoritmo de Chen e Huang (2016) pode ser vista na Seção 2.3. Cada experimento foi realizado com e sem a inclusão de instrução especialista e possui as seguintes tarefas:

- treinamento do algoritmo com 1000, 4000, 7000, 10000 e 13000 dados;
- teste do algoritmo com dados diferentes, em quantidade igual a 1000, 4000, 7000, 10000 e 7000 dados;
- uso de IE estimada com base na IE original, conforme explicado na Seção 3.3;
- cálculo da acurácia, conforme explicado na Seção 1.2.2.

Todos os experimentos foram realizados duas vezes com dados selecionados aleatoriamente, não foram realizadas mais repetições dos experimentos por causa do tempo disponível para experimentação. Em seguida foram gerados gráficos contendo as acurácias médias obtidas nesses experimentos. Os dados utilizados no experimento são imagens de semáforos retirados de imagens do trânsito obtidas do interior de um veículo.

### 3.1.2 Experimento 2

Também foi realizado um teste incluindo a instrução especialista no fluxo completo de detecção e reconhecimento de semáforos. O algoritmo de detecção de semáforos de Chen e Huang (2016) foi implementado e incluído no experimento. As regiões selecionadas pelo



algoritmo de detecção foram submetidas à classificação pelo algoritmo de Chen e Huang (2016) treinado com instrução especialista, e depois sem instrução especialista. As regiões identificadas como semáforos foram marcadas na imagem de teste original para avaliação de um especialista humano. Após a avaliação foram calculadas as taxas de precisão e cobertura. Foram utilizados neste experimento as 17 sequências de teste do dataset de Chen e Huang (2016). As atividades realizadas no teste estão listadas a seguir:

- treinamento do algoritmo com 7000 dados;
- uso do algoritmo de detecção de Chen e Huang (2016) para encontrar ROIs em cenas de trânsito obtidas do interior de um veículo;
- teste do algoritmo com 17 sequências de cenas de trânsito;
- uso da IE original, conforme explicado na Seção 3.4;
- cálculo de precisão e de cobertura, conforme explicação apresentada na Seção 1.2.2.

As Seções 3.2 e 3.3 detalham o Experimento 1, mostrando em que foi baseada a instrução especialista utilizada neste trabalho e como ela foi incorporada ao algoritmo de aprendizado de máquina de Chen e Huang (2016), respectivamente. Em seguida, a Seção 3.4 mostra como o treinamento foi utilizado no Experimento 2 definido acima.

## 3.2 Instrução Especialista

O conhecimento especialista escolhido para ser utilizado como instrução foi a localização do semáforo numa dada imagem obtida do interior do veículo. A ideia por trás disso é que há maior probabilidade do semáforo aparecer em certas regiões como, por exemplo, nas regiões centrais e superiores da imagem, uma vez que os semáforos sempre se encontram suspensos em postes e precisam ser visíveis pelo motorista à distância.

Para testar essa ideia, mapas de frequência de semáforos foram construídos. Na construção, um especialista humano analisa uma série de imagens marcando em cada imagem a região em que o(s) semáforo(s) aparece(m). Uma matriz de mesma dimensionalidade das imagens analisadas é criada, as posições da matriz equivalentes às regiões marcadas pelo especialista humano recebem um incremento a cada imagem marcada. Ao final, toda a matriz é dividida pelo total de imagens analisadas, gerando assim as frequências de semáforos para cada pixel da imagem.

Os datasets construídos por Shi, Zou e Zhang (2016) e por Chen e Huang (2016) foram utilizados para gerar dois mapas de frequência. Uma amostra aleatória de 650 imagens foi gerada a partir dos dados de teste de cada um desses datasets, totalizando um total de 1300 imagens analisadas. Nas Figuras 6 e 7 podemos observar a representação gráfica dos mapas de frequência

obtidos dos datasets de Shi, Zou e Zhang (2016) e de Chen e Huang (2016), respectivamente; na parte inferior se encontra a escala de cores utilizada, sendo o menor valor o extremo esquerdo e o maior o extremo direito. Os dados de frequência foram suavizados utilizando um algoritmo de média com máscara 7.

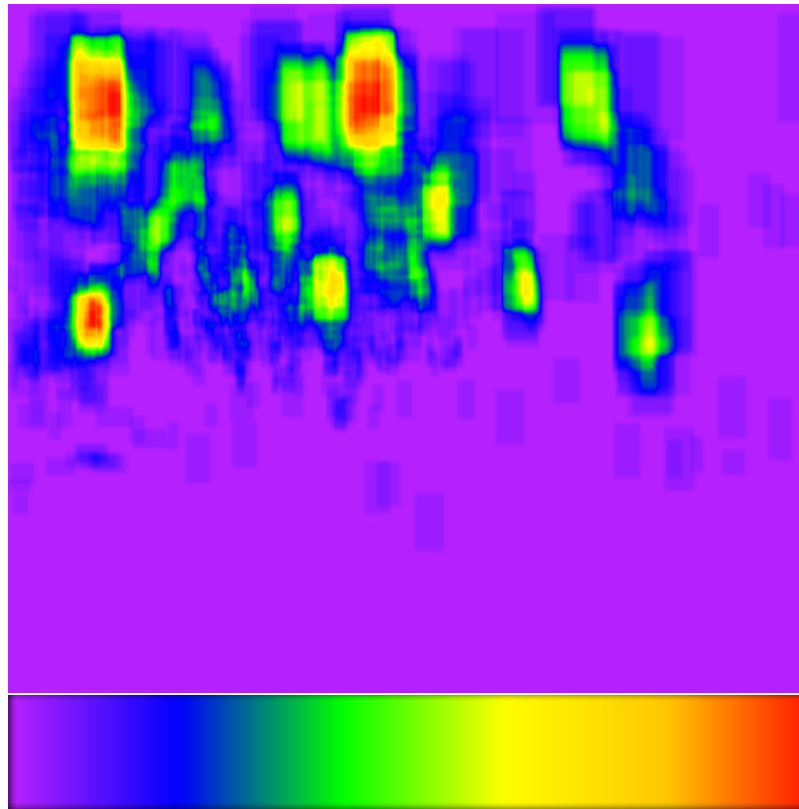


Figura 6 – Mapa de frequência de semáforos obtido a partir da amostra aleatória do dataset criado por Shi, Zou e Zhang (2016)

Para comparar os resultados de um algoritmo que utilize o conhecimento especialista e um algoritmo que não o utilize é necessário treinar os dois algoritmos com os mesmos dados. Porém os datasets encontrados na literatura não disponibilizam as coordenadas de onde as ocorrências dos semáforos foram obtidas das imagens originais, impossibilitando a combinação direta entre a frequência calculada anteriormente e os exemplares de treinamento.

Para resolver esse problema, foi assumido que todos os exemplares de treinamento foram encontrados em regiões de valor diferente de zero no mapa de frequência, assim a combinação poderia ser feita com valores de frequência aleatórios. Verificou-se que os valores de frequência seguem a distribuição beta, que foi utilizada para gerar os valores de frequência aleatórios de forma que ainda mantivessem relação com os valores encontrados no mapa de frequência calculado. Essa relação precisou ser mantida pois os valores originais do mapa de frequência são utilizados no Experimento 2 descrito na Seção 3.4. Nas Figuras 8 e 9 observa-se os histogramas de distribuição de frequências dos mapas exibidos nas Figuras 6 e 7, respectivamente. O eixo  $x$  representa os valores da frequência de semáforos calculados anteriormente, e o eixo  $y$  representa a frequência desses valores.

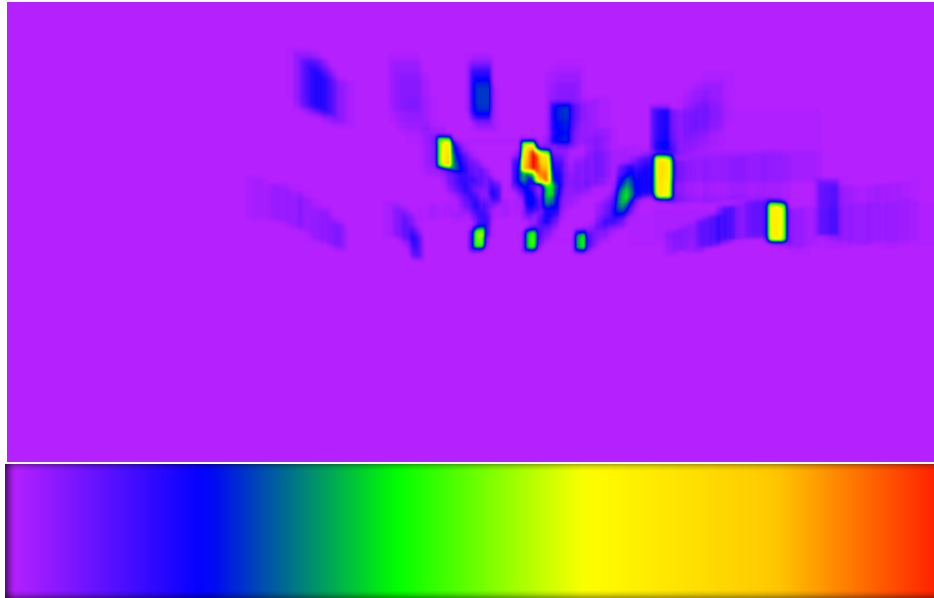


Figura 7 – Mapa de frequência de semáforos obtido a partir da amostra aleatória do dataset criado por Chen e Huang (2016)

Os valores dos parâmetros da distribuição beta encontrados para o mapa da Figura 6 foram ( $\alpha = 0.01655, \beta = 1$ ), e para o mapa da Figura 7 foram ( $\alpha = 0.03420, \beta = 12.8985$ ). Estes parâmetros foram utilizados para estimar os valores de frequência aleatórios comentados no parágrafo anterior.

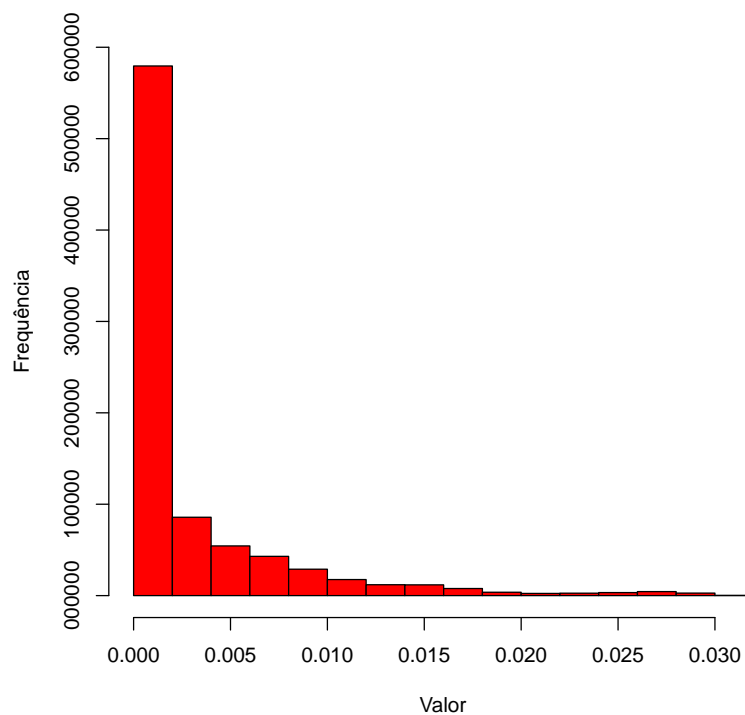


Figura 8 – Histograma da distribuição das frequências do mapa da Figura 6

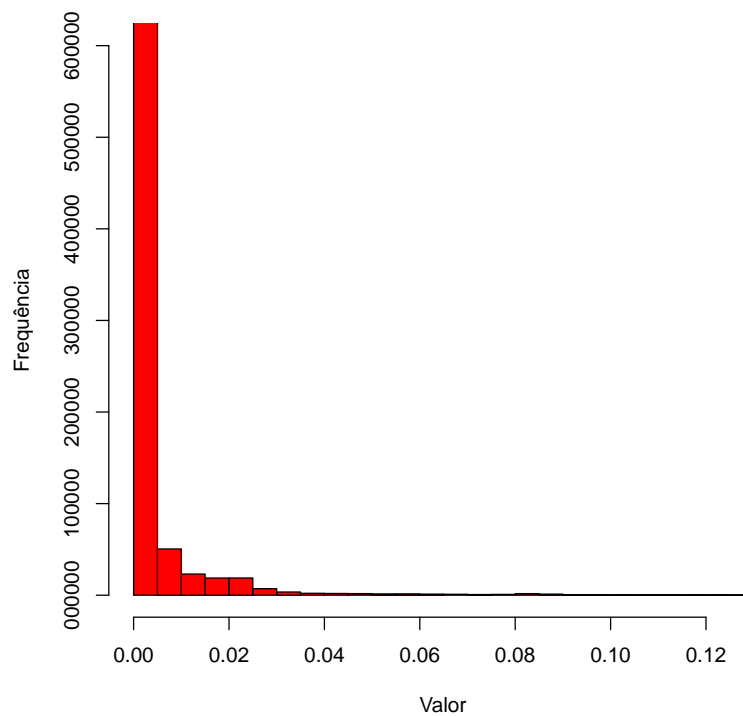


Figura 9 – Histograma da distribuição das frequências do mapa da Figura 7

### 3.3 Treinamento com Instrução Especialista

O treinamento de um algoritmo de aprendizado de máquina se dá por meio da inserção de dados no algoritmo por um número de vezes que seja suficiente para o algoritmo encontrar um padrão nesses dados. Uma quantidade razoável de dados variados são necessários nesse procedimento.

O método de reconhecimento de semáforos utilizado para testar a inserção do conhecimento especialista foi o proposto por Chen e Huang (2016). Os autores propuseram uma PCANet para encontrar vetores de características dado o conjunto de semáforos informados para treinamento. Os vetores de características são então utilizados para treinar SVMs que classificarão os pixels como pertencentes ou não a um semáforo.

A inclusão da instrução especialista nesse processo pode se dar de várias formas. É possível, por exemplo, obter as coordenadas da cena original da região classificada pelo método como semáforo e analisar o mapa de frequência nas mesmas coordenadas verificando se os valores estão acima de um determinado limiar para ser então considerado um semáforo. O problema dessa alternativa é que o algoritmo se torna demasiadamente determinístico, uma vez que apenas as regiões estáticas em que aparecem os semáforos no mapa de frequência obteriam resultados positivos.

Outra forma de uso da instrução especialista seria realizar uma operação de soma ou

multiplicação em uma das fases internas do método de reconhecimento. Uma dessas fases poderia ser entre as camadas da PCANet, ou na saída da PCANet antes da entrada de dados na SVM. O problema da combinação nestes momentos nos algoritmos citados é o tamanho da imagem que vai diminuindo com as convoluções no interior do algoritmo, dificultando a combinação direta com o valor do mapa de frequências.

A forma mais adequada definida para o caso abordado nesse trabalho foi tratar a instrução especialista como a primeira camada do método de reconhecimento. Dessa forma, após a fase de detecção, as regiões de interesse (Regions of Interest - ROI) são somadas ou multiplicadas pelo valor da frequência obtida para aquela região no mapa de frequências. Observando que o mapa de frequências e a cena original possuem a mesma dimensionalidade, e que se está de posse das coordenadas do ROI na cena original, essa combinação tenderá a destacar o ROI encontrado para o algoritmo seguinte. A operação utilizada neste trabalho foi a multiplicação. Na Figura 10 observa-se a estrutura de uma PCANet em forma de modelo de processos. É possível perceber a inclusão da instrução especialista logo após a obtenção da imagem de entrada para classificação, nesse caso, os ROIs. Esta inclusão de IE no algoritmo de aprendizado de máquina foi a principal contribuição do trabalho.

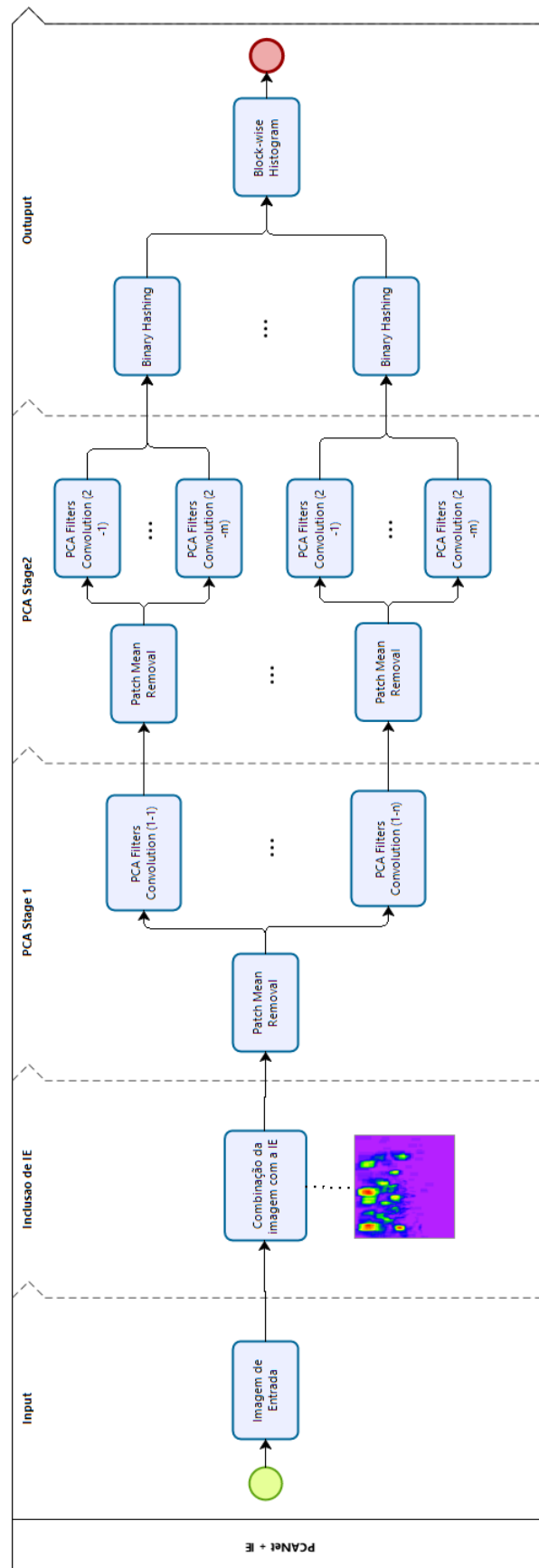


Figura 10 – Estrutura de uma PCANet de duas fases (Stages) com inclusão de IE.

Uma preocupação importante neste momento do trabalho é garantir que a instrução especialista não torne o método determinístico, como aconteceria se ela fosse aplicada no final do método. É possível que alguma parte do mapa de frequência possua valor 0, o que significa que na amostra analisada nenhum semáforo apareceu naquela região. Entretanto, embora as chances sejam pequenas, é também possível que um semáforo apareça em uma região não comum da cena, e o DRS precisa estar habilitado a encontrá-lo.

Para evitar que a combinação da instrução especialista com os dados acabe por anular completamente algum dado foi necessário definir um incremento *inc* que é adicionado à todo fator de multiplicação *fat*, que consiste no valor da frequência encontrada no mapa de frequência. Em todos os experimentos retratados nesse capítulo neste capítulo foi utilizado o valor  $inc = 0,1$ .

O *dataset* de treinamento de Shi, Zou e Zhang (2016) é composto de 2 classes principais: VERDE e VERMELHO. Cada uma dessas classes possui dados que podem ser distribuídos em 4 classes de acordo com o tipo do semáforo: redondo, com seta para cima, com seta para esquerda, e com seta para a direita. Nesta fase do trabalho foi utilizada apenas a divisão principal entre VERDE e VERMELHO. O *dataset* possui 9.977 exemplares de semáforos verdes e 10.975 exemplares de semáforos vermelhos, somando um total de 20.952 exemplares de treinamento.

O método de reconhecimento proposto por Chen e Huang (2016) foi treinado com diferentes quantidades de dados obtidos do *dataset* de treinamento de Shi, Zou e Zhang (2016) sem o conhecimento especialista. Em seguida o método foi treinado com os mesmos dados nas mesmas quantidades incluindo a instrução especialista.

Após cada treinamento foi realizado um teste de acurácia utilizando a mesma quantidade de dados do treinamento, ou com a quantidade total de dados restantes não utilizados no treinamento quando a mesma quantidade não estava disponível. As quantidades de dados utilizados no treinamento foram: 1000, 4000, 7000, 10.000 e 13.000. Os treinamentos foram realizados duas vezes para cada quantidade de dados e a média das acurácias obtidas nos dois testes foi calculada.

Na Figura 11 pode-se observar as acurácias encontradas pelos sucessivos testes. Percebe-se que quando o treinamento se deu com 1000, 4000 e 7000 exemplares o treinamento com instrução especialista (linha laranja) obteve taxas de acurácia bastante superiores ao do treinamento sem instrução especialista (linha azul), com diferença de mais de 15% de acurácia. As taxas só se aproximaram quando a quantidade de dados de treinamento subiu para 10.000 exemplares, quando a instrução especialista (IE) alcançou 98,38% de acurácia e o método sem instrução especialista obteve 93,02% de acurácia. Quando o treinamento foi realizado com 13.000 exemplares as taxas se aproximaram ainda mais, a técnica com IE alcançou 85,02% e sem IE obteve 86,48%. Neste último teste percebe-se que, embora o teste sem IE tenha obtido uma acurácia maior, a diferença é de apenas 1,46%, que não é estatisticamente significativa (p-valor do teste para diferença de proporções igual a 0,8092). Nota-se também uma queda nas acurácias dos dois testes, comportamento característico de quando acontece o *overfitting*, ou seja uma alta

especialização do algoritmo aos dados.

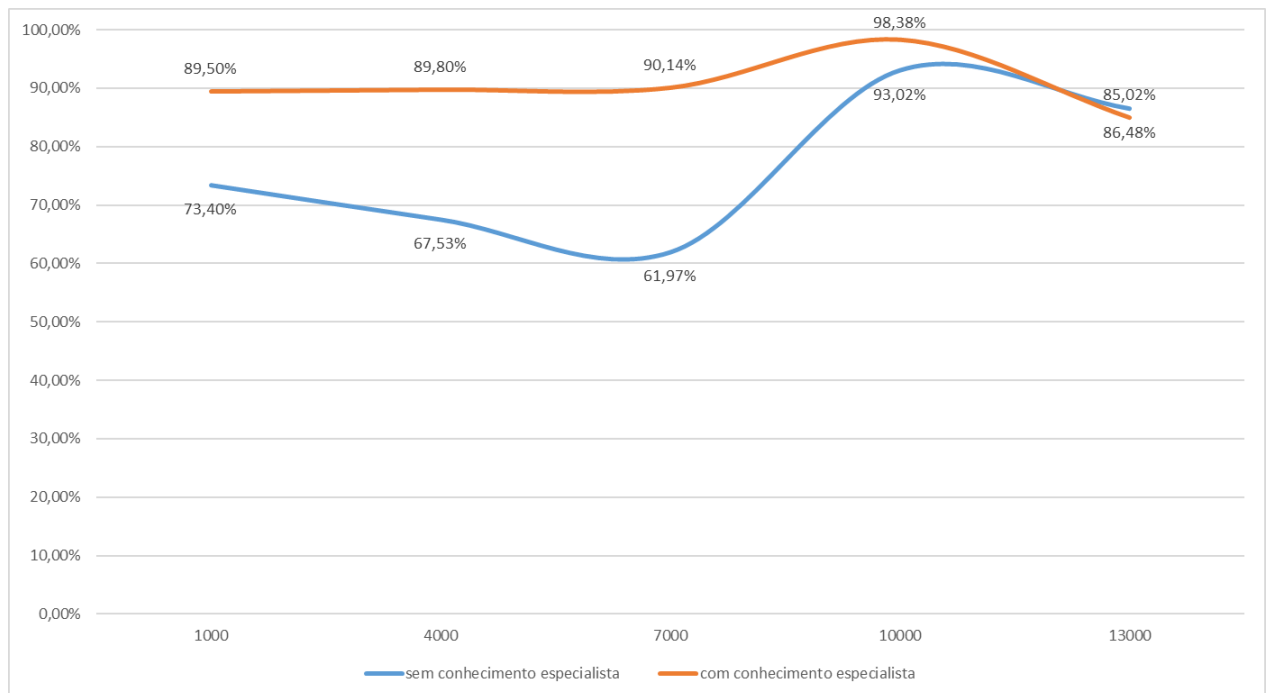


Figura 11 – Acurácias de teste com o banco de Shi, Zou e Zhang (2016) após o treinamento com IE (linha laranja) e sem IE (linha azul)

O mesmo experimento foi realizado com o *dataset* disponibilizado por Chen e Huang (2016), utilizando as mesmas quantidades de dados de treinamento. Assim como no experimento anterior, cada uma das classes desse *dataset* possui dados que podem ser distribuídos em 4 classes de acordo com o tipo do semáforo: redondo, com seta para cima, com seta para esquerda, e com seta para a direita. Porém, neste caso, os dados estão divididos em grupos mais específicos a depender de como o semáforo se apresenta, sendo estes: GreenRoi1, GreenRoi3 e GreenRoi4 para variações de semáforos verdes; e RedRoi1 e RedRoi3 para variações de semáforos vermelhos.

Na Figura 12 observam-se as médias das taxas de acurácia encontradas pelos testes no *dataset* de Chen e Huang (2016), em cada grupo de semáforos. Percebe-se que, assim como nos resultados apresentados na Figura 11, quando o treinamento se deu com 1000, 4000 e 7000 o uso do Instrução Especialista (IE) aumentou a acurácia em pelo menos 16,07% com relação ao treinamento convencional. Os testes realizados com o conhecimento especialista obtiveram no mínimo 75% de acurácia, mesmo utilizando a menor quantidade de dados possível nos testes: 1000 exemplares.

As taxas de acurácia aproximaram-se ao utilizar 13.000 exemplares para o treinamento, quando o uso de IE alcançou 82,51% de acurácia e foi ultrapassado pela técnica convencional que obteve 86,29% de acurácia. Neste caso, constata-se uma diferença estatisticamente significativa de mais de 4% entre as duas técnicas (p-valor do teste para diferença de proporções menor que  $2,2^{-16}$ ). Sendo assim, com uma maior quantidade de dados a técnica sem IE, nesse cenário, comportou-se melhor que usar IE, porém com pouca superioridade. Isso pode ser melhor



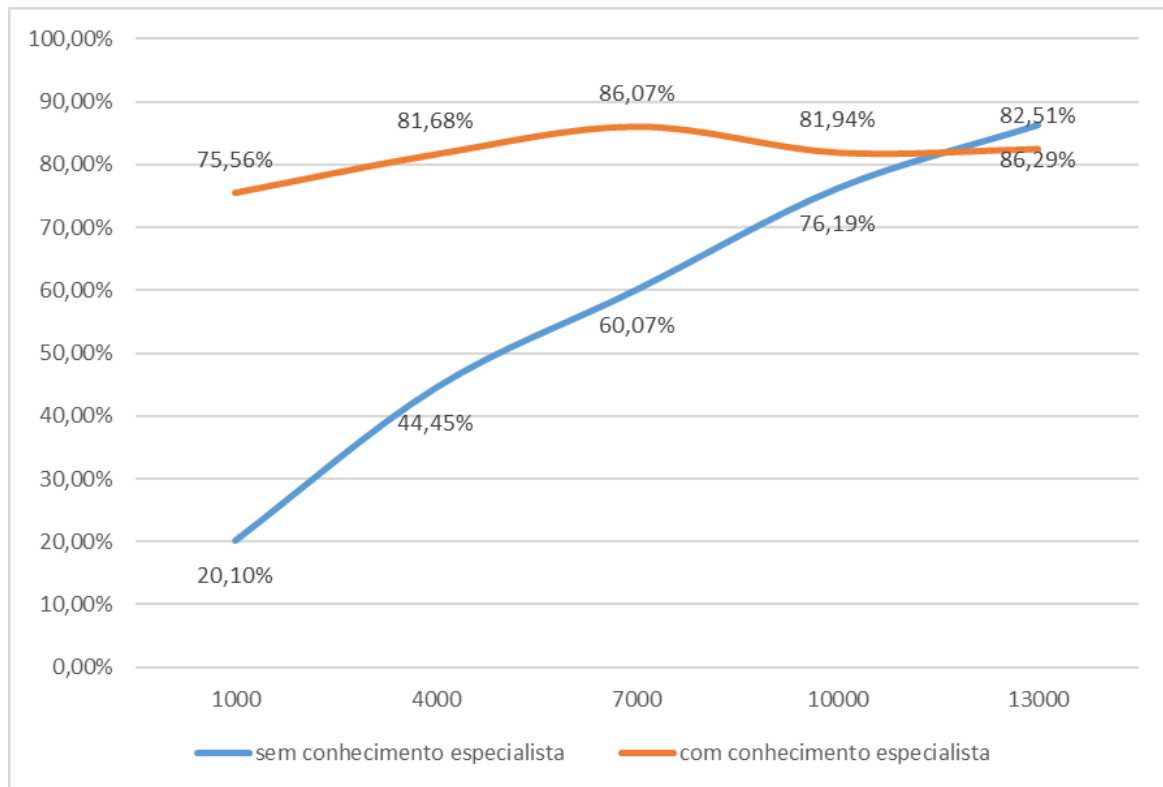


Figura 12 – Acurácias de teste com o banco de Chen e Huang (2016) após o treinamento com IE (linha laranja) e sem IE (linha azul)

compreendido ao observar a trajetória do treinamento com IE, que já começa alta e continua crescendo até alcançar os 10.000 exemplares quando começa a cair, o que parece indicar que o *overfitting* do treinamento com IE acontece mais cedo que o do treinamento sem IE.

Importante ressaltar que os resultados da aplicação de IE para diferentes dados pode ser diferente, mesmo dentro do mesmo domínio de problema. Nas Figuras 13, 14, 15, 16 e 17 observa-se o detalhe das acurácias de cada grupo de semáforos que geraram as acurácias da Figura 12. Percebe-se que, embora em todos os grupos a IE tenha melhorado as taxas, os resultados dos grupos de semáforos vermelhos, nas Figuras 16 e 17, apresenta melhora visualmente menor que a melhora observada nos grupos de semáforos verdes nas Figuras 13, 14 e 15. Essa característica indica uma menor sensibilidade dos semáforos vermelhos à inclusão de IE. Essa teoria também é reforçada ao observar a queda abrupta na taxa de acurácia nos grupos verdes das Figuras 13 e 14, o que denota mais uma vez a queda de poder de generalização dada a alta especialização aos dados, comportamento que não se repete com tamanha força nos grupos vermelhos.

### 3.4 Detecção e Reconhecimento Especialista

Com o objetivo de validar a inclusão de IE no fluxo completo de um DRS foi realizado um experimento realizando a fase de detecção de ROIs seguida da submissão dos ROIs encontrados ao algoritmo treinado. Foi utilizado nesse teste o algoritmo de detecção proposto por Chen

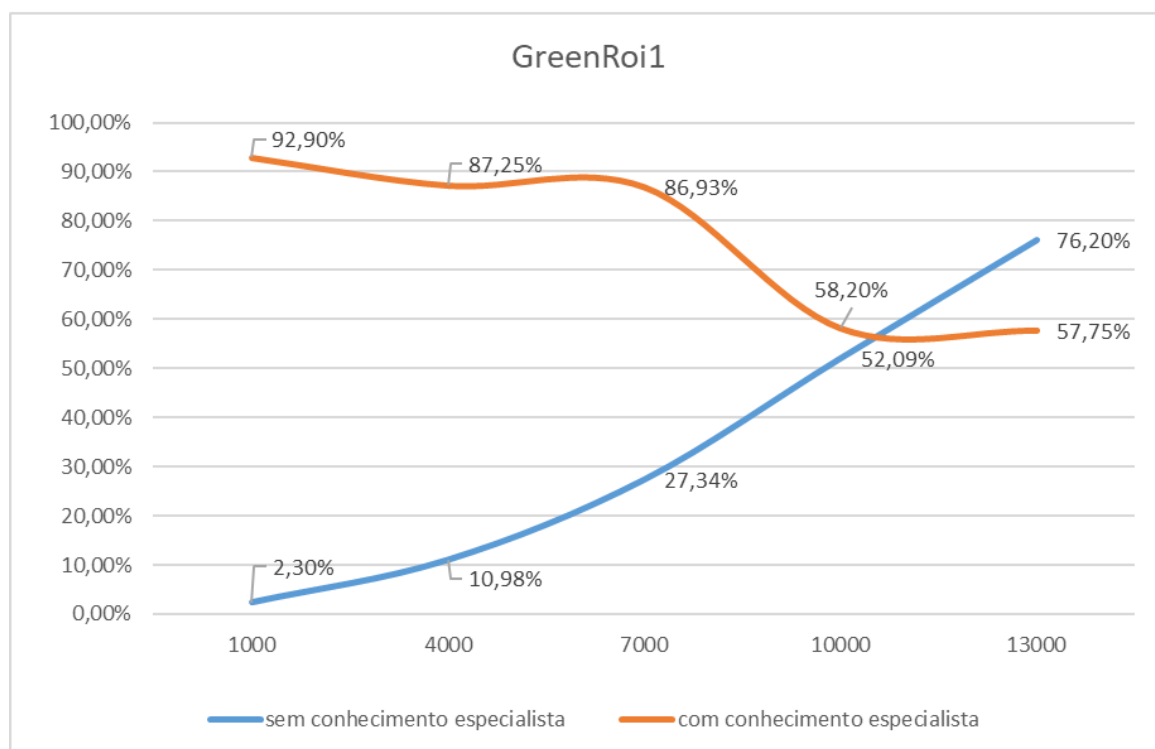


Figura 13 – Acurácias de teste do grupo de dados GREENROI1 do dataset de Chen e Huang (2016). após o treinamento com IE (linha laranja) e sem IE (linha azul)

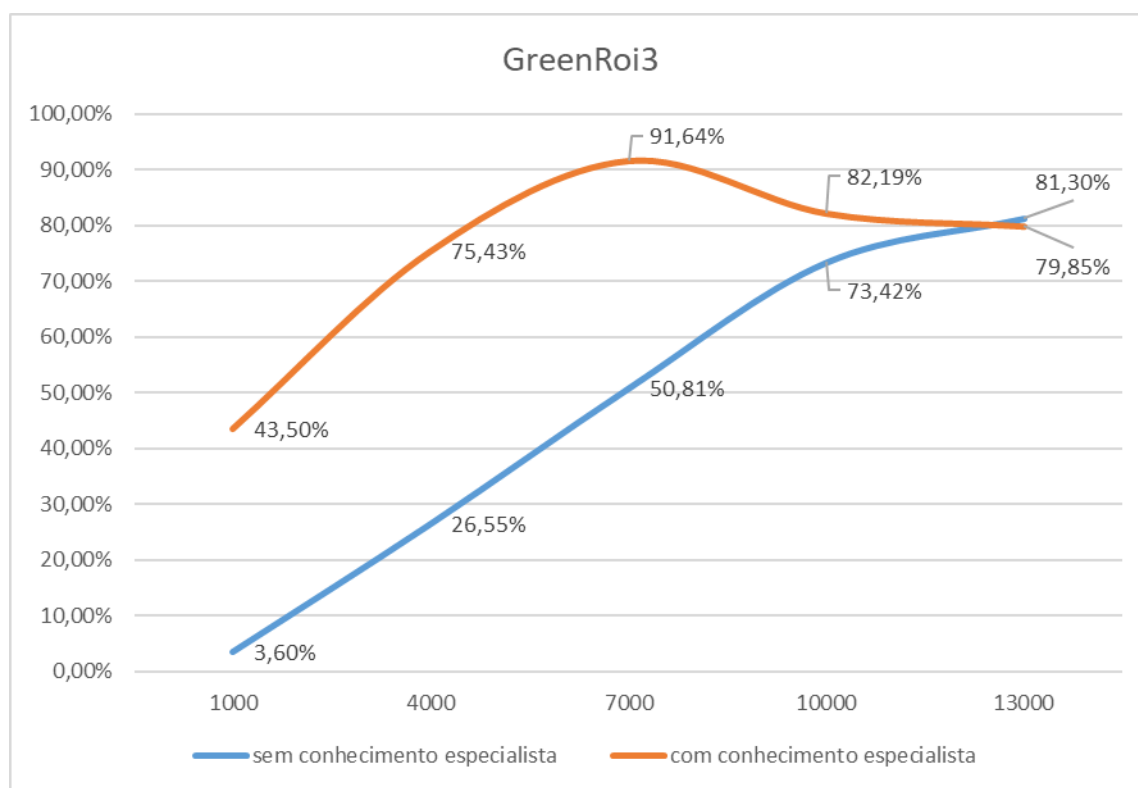


Figura 14 – Acurácias de teste do grupo de dados GREENROI3 do dataset de Chen e Huang (2016). após o treinamento com IE (linha laranja) e sem IE (linha azul)

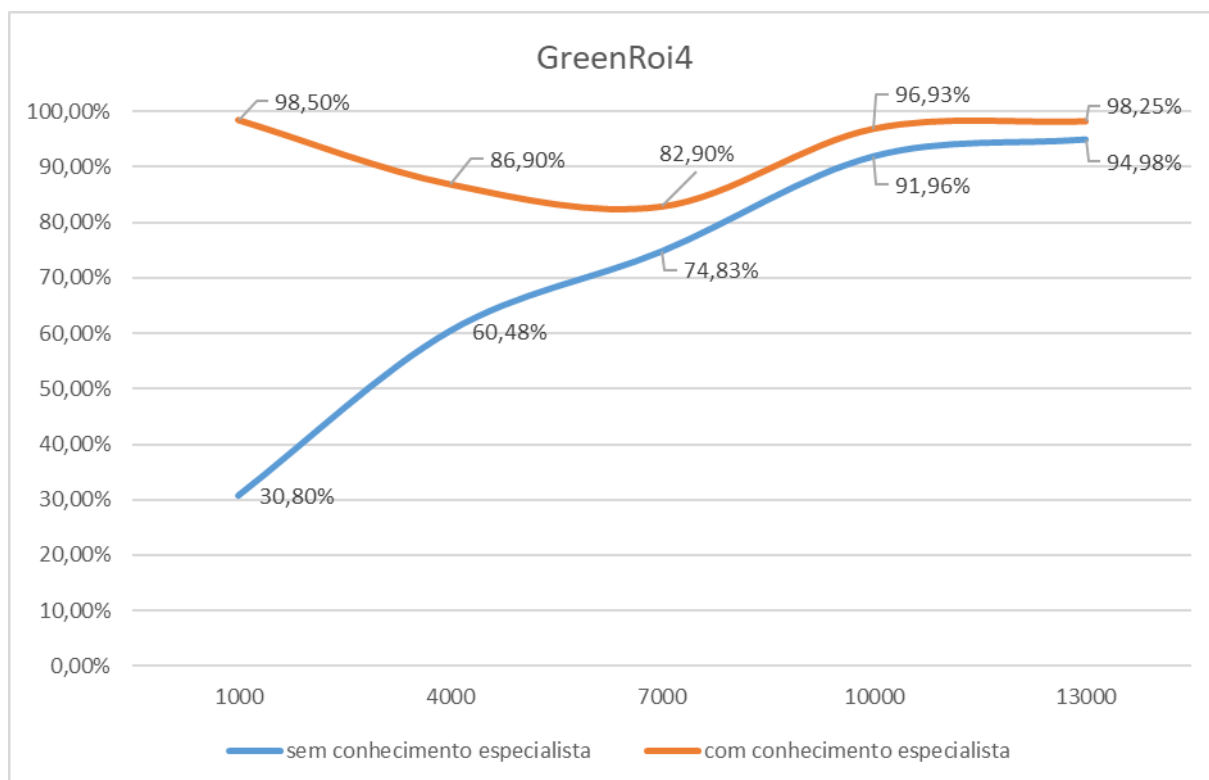


Figura 15 – Acurácias de teste do grupo de dados GREENROI4 do dataset de Chen e Huang (2016). após o treinamento com IE (linha laranja) e sem IE (linha azul)

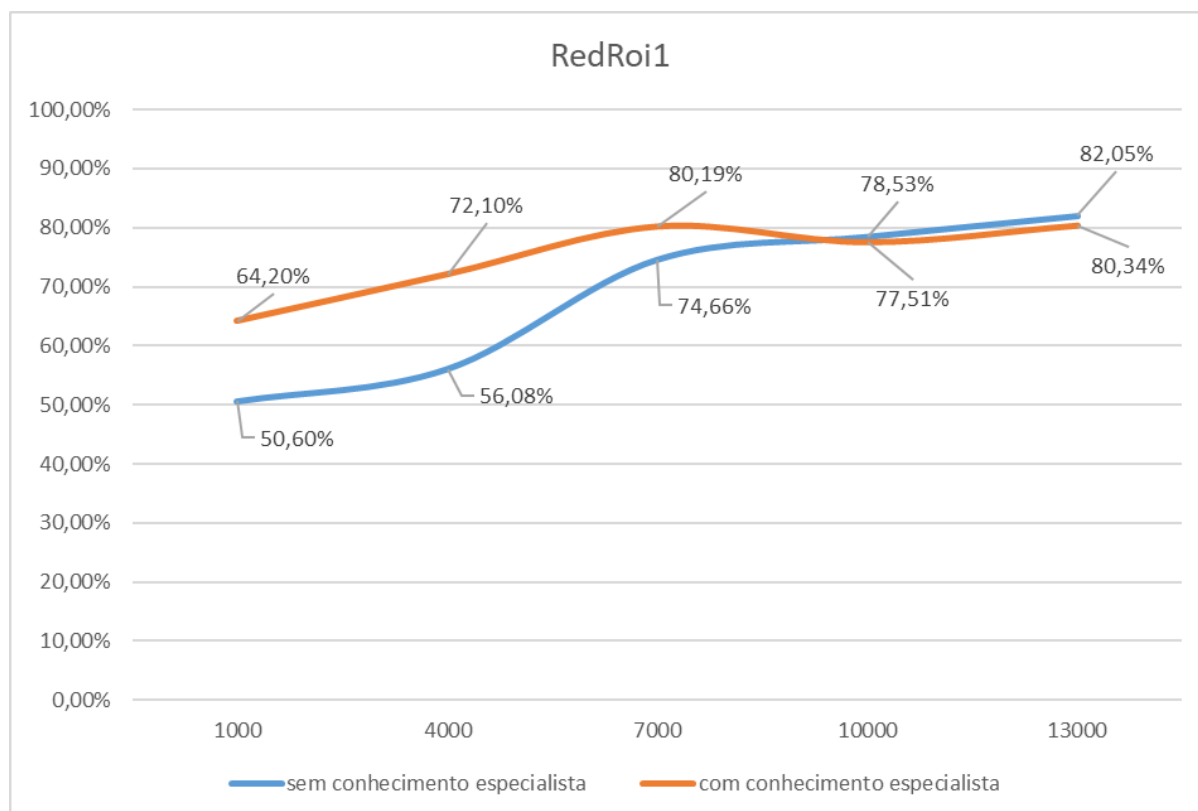


Figura 16 – Acurácias de teste do grupo de dados REDROI1 do dataset de Chen e Huang (2016). após o treinamento com IE (linha laranja) e sem IE (linha azul)

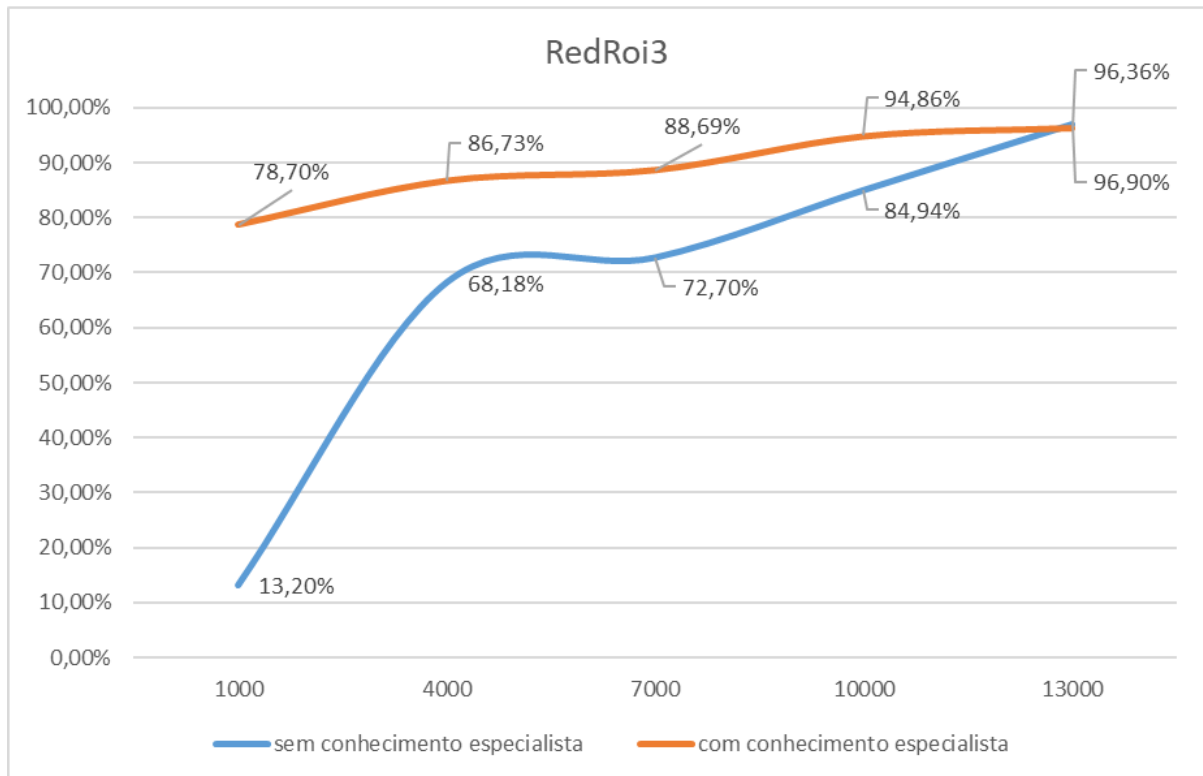


Figura 17 – Acurácias de teste do grupo de dados REDROI3 do dataset de Chen e Huang (2016). após o treinamento com IE (linha laranja) e sem IE (linha azul)

e Huang (2016) em conjunto com a PCANet do mesmo autor para classificação. A principal diferença deste experimento para o anterior é o uso do valor de frequência obtido do mapa de frequências original acrescido do incremento  $inc = 0,1$ , uma vez que agora os ROIs são detectados pelo algoritmo de detecção e guardam as coordenadas de origem da região. No experimento anterior uma estimativa de frequência era gerada baseada na distribuição das frequências originais.

No primeiro momento foi necessário realizar a reprodução do código do trabalho de Chen e Huang (2016), uma vez que este não estava disponível. O código foi implementado na linguagem Python e segue as coordenadas do trabalho de Chen e Huang (2016) apresentadas na Seção 2.3. De posse do código foi realizado um teste sem IE e um teste com IE utilizando as imagens do dataset de teste disponibilizado por Chen e Huang (2016). No dataset utilizado há 23 sequências com 8112 semáforos, porém somente 17 dessas sequências possuem semáforos e por isso foram as únicas utilizadas no teste descrito nesta seção. As 17 sequências possuem juntas 2142 imagens, a menor sequência com 56 frames, e a maior com 518.

A quantidade de dados de treinamento utilizada tanto para o teste completo com IE e sem IE foi 7000 exemplares. Esse valor foi escolhido por ser a média das quantidades de dados mínima e máxima utilizados no experimento anterior e por ter conseguido taxas de acurácia acima de 70% nos treinamentos. Não foi utilizada a quantidade exata de dados de treinamento do trabalho original na reprodução pois ela não é definida no trabalho, além disso a análise dos

resultados obtidos pelo algoritmo é manual e trabalhosa, pois as imagens precisam ser verificadas uma a uma.

A automatização de contagem de acertos nesse tipo de problema é bastante complicada pois não é possível associar uma imagem a uma classe específica, uma vez que não se trata de classificar imagens e sim objetos contidos nela. A quantidade de objetos que se deseja encontrar na imagem também pode variar em cada imagem do dataset, podendo inclusive ser 0. Um arquivo de metadados poderia ser criado contendo para cada imagem uma lista de objetos esperados com suas classes, porém, sem uma revisão manual, um objeto não-esperado pode ser classificado erroneamente numa das classes esperadas e ser contabilizado como um verdadeiro positivo, quando na realidade seria um falso positivo, o que influenciaria a robustez do resultado.

A Tabela 1 apresenta as taxas de precisão e cobertura obtidas pelo teste completo com IE e sem IE. O experimento utilizando IE se mostrou superior obtendo 83% de precisão e 73% de cobertura em média, contra 75,3% de precisão e 51,1% de cobertura.

Analisando as taxas de cada sequência individualmente na Tabela 1 observa-se que, quanto à precisão, o experimento com IE obteve taxas superiores em 12 das 17 sequências, taxas iguais ao experimento sem IE em 2 das 17 sequências (6 e 4), e taxas inferiores em apenas 3 das sequências (10, 11 e 14).

Em relação à cobertura, as taxas do experimento com IE só foram inferiores na sequência 12. Neste teste a recovocação com IE foi de 60% enquanto sem IE foi de 61,3%, valores muito próximos. No mesmo experimento, entretanto, a precisão foi de 89% para o uso de IE e 81% sem o uso de IE.

Percebemos que embora o uso de IE no tenha sido superior ao algoritmo sem uso de IE ainda há uma deficiência considerável com relação à cobertura. Isso indica que, embora o uso de IE tenha coberto 21,9% a mais do universo de semáforos existentes que o algoritmo tradicional, 27% dos ROIs que eram semáforos foram classificados como não sendo semáforo, o que pode ser muito problemático no mundo real causando acidentes graves.

Tabela 1 – Taxas de precisão e cobertura do experimento utilizando as imagens completas.

Sequência	Prec. Especialista	Cob. Especialista	Prec. PCA	Cob. PCA
1	66%	52%	56,7%	45,8%
2	100%	92%	92,0%	41,0%
3	95%	95%	78,3%	42,0%
4	95%	64%	95,0%	77,8%
5	100%	100%	89%	77,1%
6	60,4%	45%	60,1%	14,0%
7	100%	92%	89,7%	80,0%
8	98%	98%	79,7%	81,3%
9	84%	43%	76,3%	42,0%
10	79%	90%	84,3%	67,5%
11	51%	37%	62,5%	18,1%
12	89%	60%	81,8%	61,3%
13	91%	97%	63,6%	35,0%
14	92%	85%	95,0%	76,0%
15	96%	72%	84,0%	41,1%
16	87%	78%	73,5%	38,0%
17	53%	48%	48,0%	31,2%
Média	83%	73%	75,3%	51,1%

# 4

## **Protótipo de um Dispositivo de Detecção e Reconhecimento de Semáforos**

Um passo muito importantes deste trabalho foi criar um dispositivo capaz de interagir com o motorista e analisar o ambiente a partir do interior do veículo. Esse dispositivo seria então responsável pela captura de imagens para Detecção e Reconhecimento de Semáforos (DRS), por isso chamado de dispositivo DRS ou TLR - Traffic Light Recognition. O dispositivo também pode ser usado para captura de imagens para criação de um dataset local.

Este capítulo descreve a criação deste dispositivo, o seu uso para criação de um dataset local, e a aplicação do algoritmo desenvolvido por Shi, Zou e Zhang (2016) a esse dataset.

Uma das principais questões ao prototipar um Dispositivo DRS é onde posicioná-lo no veículo, uma vez que o dispositivo precisa ter uma visão clara do tráfego sem comprometer a visão do condutor. Outra preocupação importante é que o dispositivo necessita de proteção contra condições meteorológicas diversas como chuva e calor, deste modo o dispositivo precisa estar no interior do veículo ou ser à prova d'água, e sua localização precisa minimizar a incidência de luz solar direta.

Uma vez que é natural a trepidação durante a movimentação do veículo, o dispositivo precisa estar o mais estabilizado possível, de forma que a trepidação não influencie negativamente a captura de imagens e consequentemente a identificação dos semáforos.

O dispositivo também precisa ser capaz de alertar o condutor de forma não intrusiva, neste caso usando um aviso sonoro, de visualizar o tráfego por meio de uma câmera, e possuir uma unidade de processamento computacional para processar os dados obtidos. Um dispositivo acessível a boa parte da população que atende esses requisitos e que já é comumente utilizado para assistir motoristas no trânsito é o smartphone.

Um smartphone foi posicionado no interior de um veículo para capturar imagens reais do trânsito com e sem semáforos. Dois tipos de suporte são usados em geral para posicionar smartphones em posições úteis para os condutores: suportes de ar-condicionado e suporte com

ventosas para o pára-brisas. O suporte de ar-condicionado não pode ser utilizado pois não possui visão do exterior do veículo. Suportes com ventosas para pára-brisas são uma opção mais plausível, porém o suporte pode se soltar ou trepidar bastante a depender da qualidade do suporte utilizado.

Para atender os requisitos especificados nos parágrafos anteriores e resolver os problemas dos suportes apresentados um suporte estável foi desenvolvido utilizando uma fita adesiva dupla-face e parte de um suporte com ventosas. A parte do suporte com ventosas que segura o dispositivo foi separada da parte com ventosas. Então a primeira parte foi fixada na parte central do painel de um automóvel com uma fita adesiva dupla. Este design permite que o dispositivo use a camera no modo paisagem, minimizando a quantidade de céu capturada e maximizando o tamanho da cena do trânsito e consequentemente a possibilidade de encontrar semáforos (Figura 18).



Figura 18 – iPhone 6 em suporte para dispositivo DRS.

Foram utilizados três diferentes smartphones para capturar vídeos do trânsito contendo semáforos: Motorola G 2a Geração, iPhone 6, e Galaxy S8+. Todos os dispositivos foram configurados para capturar vídeo com resolução HD. A Figura 19 mostra um exemplo de imagens capturadas com estes dispositivos. As imagens foram extraídas dos vídeos numa taxa de 5 frames por segundo (fps).

Este capítulo da dissertação foi publicado com o título *A Traffic Light Recognition Device* como capítulo do livro *Information Technology - New Generations. Advances in Intelligent Systems and Computing*, vol 738. Springer.

## 4.1 Resultados Obtidos com o Protótipo

As imagens obtidas com o Dispositivo DRS construído foram submetidas a classificação em um notebook utilizando o método desenvolvido por Shi, Zou e Zhang (2016).

As imagens foram obtidas utilizando três smartphones diferentes. O primeiro grupo obtido com um Motorola G2 não apresentou bons resultados, por este motivo foi descartado



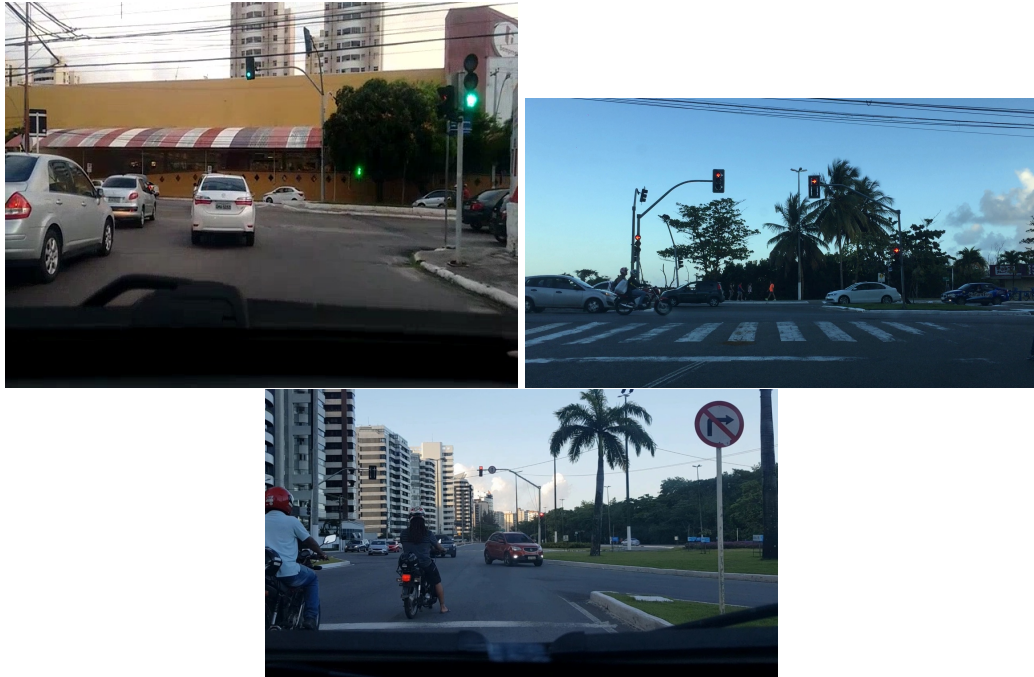


Figura 19 – Imagens obtidas utilizando o suporte para DRS prototipado com diferentes smartphones. De cima para baixo, em sentido horário: imagem obtida com um Motorola G 2ª Geração, com um iPhone 6, e com um Galaxy S8+.

e não foi contabilizado nos resultados. O segundo grupo, com imagens obtidas pelo iPhone 6, contém 682 imagens, sendo 209 sem semáforos e 473 com semáforos. O terceiro grupo é formado por 247 imagens obtidas com o Galaxy S8+, sendo 165 com semáforos e 82 sem semáforos.

Um total de 929 imagens de trânsito foi analisado: 638 imagens contendo semáforos verdes ou vermelhos, e 291 imagens que não contém semáforos, o grupo negativo.

Constatou-se também que apenas a categorização das imagens do dataset não é suficiente para a avaliação dos resultados obtidos. Suponha que o label da imagem 01 diga que há um semáforo vermelho na imagem e o DRS, diga também que há um semáforo vermelho na imagem. Suponha também que o semáforo vermelho encontrado pelo DRS é na verdade uma placa de Pare. Se a comparação for feita apenas entre os labels encontrados o resultado será 1 verdadeiro positivo para a imagem 01, quando na verdade deveria ser 1 falso positivo. Sendo assim, ao se trabalhar com reconhecimento de objetos isolados na imagem como um semáforo é necessário que o DRS destaque na imagem obtida a região identificada como semáforo, e ao computar os resultados o pesquisador deve verificar cada imagem uma a uma.

Considerando que na maioria das vezes há dois semáforos iguais para a mesma via foi contabilizado apenas um resultado positivo para o tipo de semáforo encontrado quando um ou ambos os semáforos são reconhecidos na imagem. Sendo assim para cada imagem nós temos 01 erro ou 01 acerto. Isso reflete o comportamento no dia a dia quando é necessário observar apenas um semáforo para fazer uma escolha.

Na Figura 20 é possível observar os resultados detalhados de obtidos por cada grupo de imagens. Os dois grupos alcançaram altas taxas de precisão, porém o grupo de imagens obtidas com o iPhone 6 apresentou uma taxa de cobertura de apenas 60%. Este resultado pode ser explicado pelo fato de que os exemplares de semáforos utilizados no treinamento são muito diferentes de alguns dos exemplares encontrados no grupo de teste, característica demonstrada na Figura 21. Quando os exemplares de treinamento não representam de forma completa o mundo real alguns semáforos acabam por não ser reconhecidos.

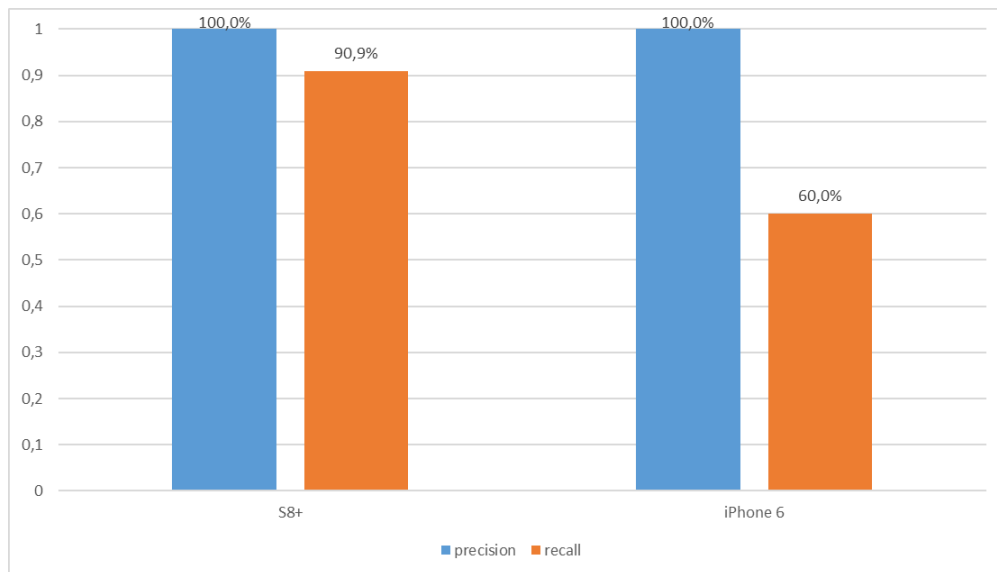


Figura 20 – Taxas de precisão e cobertura (*recall*) por smartphone utilizado para capturar as imagens.

As condições de iluminação no dataset de treinamento também são muito diferentes das condições em que o dataset de teste foi criado em consequência da região geográfica, interferência meteorológica e possivelmente diferença no dispositivo utilizado para obter as imagens dos datasets. Condições como essas podem influenciar o resultado.

É importante considerar que a distância entre o DRS e o semáforo é crucial para o reconhecimento. Quanto maior a distância a partir da qual o DRS consegue classificar corretamente o semáforo maior a sua robustez. Para esta pesquisa, foi utilizada uma distância relativa considerando para isso o tamanho do semáforo na imagem; foram somente contabilizados nos resultados apresentados os semáforos com ao menos 20 pixels na sua diagonal.

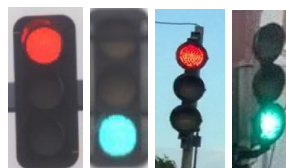


Figura 21 – Da esquerda para a direita: exemplos de semáforo verde e vermelho utilizados no treinamento, exemplos de semáforo verde e vermelho obtidos pelo DRS prototipado e utilizados nos testes.

A baixa taxa de cobertura do grupo de teste obtido com o iPhone influenciou negativamente os resultados finais dos testes com o DRS, como observado na Figura 22. Em comparação com os resultados originais de Shi, Zou e Zhang (2016) e com a nossa reprodução utilizando os seus dados, os resultados validam o DRS e justificam o seu uso em pesquisas futuras.

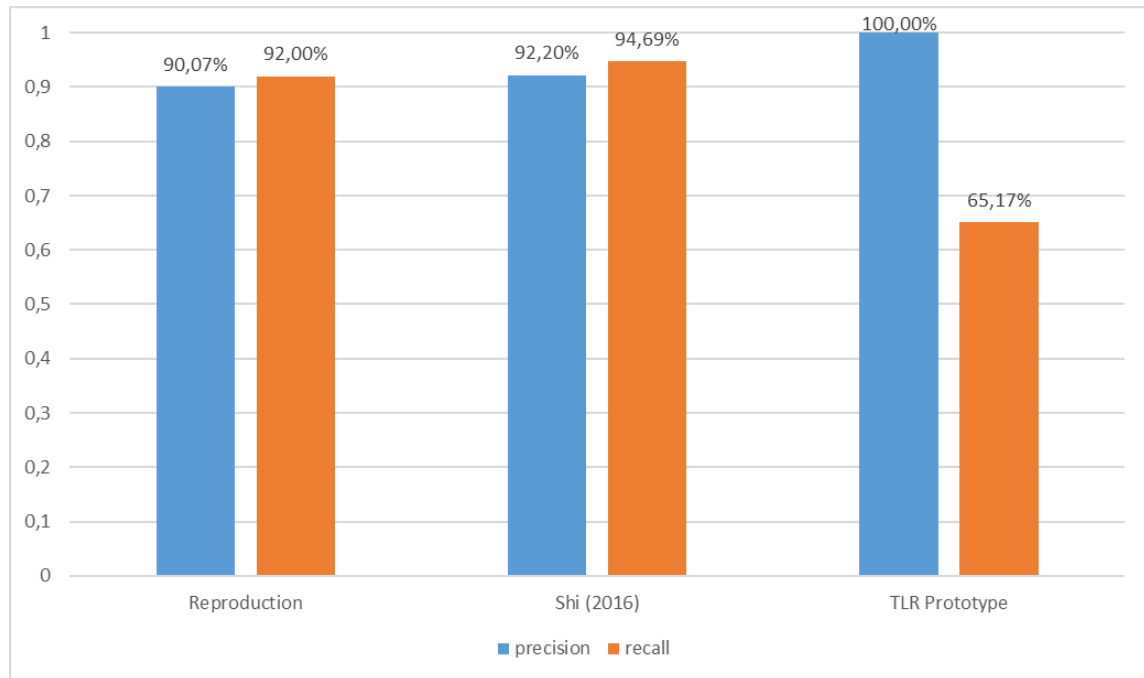


Figura 22 – Taxas de precisão e cobertura (*recall*): da nossa reprodução do trabalho de Shi, Zou e Zhang (2016), do trabalho original de Shi, Zou e Zhang (2016), e dos testes utilizando as imagens obtidas pelo DRS prototipado, respectivamente.

# 5

## Conclusão

O principal objetivo desta dissertação foi avaliar se a inclusão de Instrução Especialista (IE) poderia diminuir a quantidade de dados necessários para treinar um algoritmo de aprendizado de máquina no problema de reconhecimento de semáforos. Para atingir esse objetivo foi realizado o treinamento de uma PCANet com e sem IE. A IE utilizada foi a frequência de ocorrência de semáforo por pixel de uma sequência de imagens. A escolha da IE foi feita com base no conhecimento implícito ao condutor de esperar que o semáforo se encontre nas regiões centrais e superiores da cena que ele observa do interior do veículo. Foram construídos mapas de frequência para validar essa ideia.

Os resultados obtidos após o treinamento confirmam que o uso de IE ajuda a diminuir quantidade de dados para o treinamento, pois tanto no dataset de (SHI; ZOU; ZHANG, 2016) como no de (CHEN; HUANG, 2016), os resultados dos testes após treinamento com IE foram superiores aos do treinamento tradicional, ou seja, sem IE. Mesmo quando a quantidade de exemplares de treinamento era igual a 1000 exemplares, a menor quantidade de dados utilizada nos experimentos, o treinamento com IE conseguiu ao menos 75% de acurácia após o treinamento.

Para avaliar o uso desse treinamento num processo de reconhecimento de semáforos completo foi realizado um experimento em que um algoritmo de detecção de ROIs foi aplicado e em seguida os ROIs foram submetidos à classificação pela PCANet utilizando IE e sem utilizar IE. O experimento utilizando IE se mostrou superior obtendo 83% de precisão e 73% de cobertura em média, contra 75,3% de precisão e 51,1% de cobertura.

Um protótipo de DRS também é apresentado no Capítulo 4, o objetivo é mostrar uma forma de obter imagens para teste neste tipo de problema bem como um *layout* que pode ser utilizado na criação de dispositivos finais para auxílio do motorista. As imagens obtidas com o DRS foram submetidas ao algoritmo de detecção e reconhecimento de semáforos proposto por (SHI; ZOU; ZHANG, 2016). A precisão de 100% e cobertura de 65,17% validam o uso deste *layout* de DRS em pesquisas futuras.

Baseado nos resultados obtidos foi possível concluir que o uso de IE na área de reconhecimento de semáforos reduziu a quantidade de dados necessárias no treinamento de algoritmos de aprendizado de máquina. Os resultados promissores também validam a investigação da mesma técnica em outras áreas de conhecimento e em conjunto com outros algoritmos de aprendizado de máquina.

Importante ressaltar que os resultados de todos os experimentos ainda podem ser melhorados alterando-se os parâmetros de elaboração da base de IE apresentados no Capítulo 3. Um desses parâmetros é a quantidade de imagens utilizadas para gerar o mapa de frequências, que pode ser expandida para ser ainda mais representativo, porém é necessário um estudo para analisar se o custo de analisar manualmente mais imagens traz uma melhoria palpável aos resultados do método. Outros parâmetros que também podem ser alterados são: a máscara utilizada para suavização do mapa de frequências e o incremento *inc* adicionado a todo valor de frequência *fat* para a combinação com a imagem de entrada do algoritmo. Testes alterando os parâmetros citados serão realizados em trabalhos futuros

Outros trabalhos futuros derivados dessa dissertação podem ser: investigar tipos de bases de IE e a melhor forma de construí-los, o impacto de usar IE em diferentes algoritmos de aprendizado, o uso de IE em diferentes tipos de problemas, a criação de um aplicativo ou dispositivo que possa ser utilizado pelo público.

# Referências

AFFONSO, C. et al. Deep learning for biological image classification. v. 85, 05 2017. Citado 2 vezes nas páginas 21 e 23.

ALMAGAMBETOV, A.; VELIPASALAR, S.; BAITASSOVA, A. Mobile standards-based traffic light detection in assistive devices for individuals with color-vision deficiency. *IEEE Transactions on Intelligent Transportation Systems*, v. 16, n. 3, p. 1305–1320, June 2015. ISSN 1524-9050. Citado na página 25.

Almeida, A. B. *Usando o Computador para Processamento de Imagens Médicas*. 1998. Acesso em: 15 fev. 2015. Disponível em: <<http://www.informaticamedica.org.br/informaticamedica/n0106/imagens.htm>>. Citado na página 19.

ALMEIDA, T.; VASCONCELOS, N.; BENICASA, A. Framework para detecção de semáforos baseado em atenção visual. In: *Conference on Graphics, Patterns and Images, 28.(SIBGRAPI), Salvador*. [S.l.: s.n.], 2015. Citado 2 vezes nas páginas 24 e 25.

ALMEIDA, T. et al. Fuzzy model applied to the recognition of traffic lights signals. In: *2016 8th Euro American Conference on Telematics and Information Systems (EATIS)*. [S.l.: s.n.], 2016. p. 1–4. Citado 2 vezes nas páginas 24 e 25.

BALCEREK, J. et al. Automatic detection of traffic lights changes from red to green and car turn signals in order to improve urban traffic. In: *2014 Signal Processing: Algorithms, Architectures, Arrangements, and Applications (SPA)*. [S.l.: s.n.], 2014. p. 110–115. ISSN 2326-0262. Citado na página 25.

BORRMANN, J. M. et al. Stellar - A case-study on systematically embedding a traffic light recognition. In: *17th International IEEE Conference on Intelligent Transportation Systems (ITSC)*. [S.l.: s.n.], 2014. p. 1258–1265. ISSN 2153-0009. Citado na página 24.

BRAGA, A. de P. *Redes neurais artificiais: teoria e aplicações*. LTC Editora, 2007. ISBN 9788521615644. Disponível em: <<https://books.google.com.br/books?id=R-p1GwAACAAJ>>. Citado 2 vezes nas páginas 20 e 21.

CAI, Z.; LI, Y.; GU, M. Real-time recognition system of traffic light in urban environment. In: *2012 IEEE Symposium on Computational Intelligence for Security and Defence Applications*. [S.l.: s.n.], 2012. p. 1–6. ISSN 2329-6267. Citado na página 25.

CARVALHO, A.; FACELI, K. *Inteligência artificial: uma abordagem de aprendizado de máquina*. Grupo Gen - LTC, 2011. ISBN 9788521618805. Disponível em: <<https://books.google.com.br/books?id=4DwelAEACAAJ>>. Citado 3 vezes nas páginas 19, 20 e 22.

CET - Companhia de Engenharia de Tráfego. *Análise da mortalidade ocorrida por acidentes de trânsito em agosto e setembro de 2016*. 2016. Acesso em: 09 ago. 2017. Disponível em: <<http://www.cetsp.com.br/media/523410/parcial2016.pdf>>. Citado na página 10.

CHARETTE, R. de; NASHASHIBI, F. Real time visual traffic lights recognition based on spot light detection and adaptive traffic lights templates. In: *2009 IEEE Intelligent Vehicles Symposium*. [S.l.: s.n.], 2009. p. 358–363. ISSN 1931-0587. Citado 3 vezes nas páginas 25, 27 e 29.

CHEN, Q.; SHI, Z.; ZOU, Z. Robust and real-time traffic light recognition based on hierarchical vision architecture. In: *2014 7th International Congress on Image and Signal Processing*. [S.l.: s.n.], 2014. p. 114–119. Citado na página 25.

CHEN, Z.; HUANG, X. Accurate and reliable detection of traffic lights using multiclass learning and multiobject tracking. *IEEE Intelligent Transportation Systems Magazine*, v. 8, n. 4, p. 28–42, winter 2016. ISSN 1939-1390. Citado 18 vezes nas páginas 7, 24, 25, 26, 27, 29, 30, 31, 32, 33, 34, 37, 38, 39, 40, 41, 42 e 50.

CHIU, Y. T.; CHEN, D. Y.; HSIEH, J. W. Real-time traffic light detection on resource-limited mobile platform. In: *2014 IEEE International Conference on Consumer Electronics - Taiwan*. [S.l.: s.n.], 2014. p. 211–212. Citado na página 24.

CHOI, J.; AHN, B. T.; KWEON, I. S. Crosswalk and traffic light detection via integral framework. In: *The 19th Korea-Japan Joint Workshop on Frontiers of Computer Vision*. [S.l.: s.n.], 2013. p. 309–312. Citado na página 25.

CHUNG, Y.-C.; WANG, J.-M.; CHEN, S.-W. A vision-based traffic light detection system at intersections. *Journal of National Taiwan Normal University: Mathematics, Science & Technology*, v. 47, 2002. Citado 2 vezes nas páginas 25 e 27.

DIAZ-CABRERA, M.; CERRI, P.; SANCHEZ-MEDINA, J. Suspended traffic lights detection and distance estimation using color features. In: *2012 15th International IEEE Conference on Intelligent Transportation Systems*. [S.l.: s.n.], 2012. p. 1315–1320. ISSN 2153-0009. Citado na página 11.

FACON, J. Processamento e análise de imagens. PUCP, 2002. Citado 2 vezes nas páginas 16 e 17.

FAIRFIELD, N.; URMSON, C. Traffic light mapping and detection. In: *Proceedings of ICRA 2011*. [S.l.: s.n.], 2011. Citado na página 29.

FAN, B.; LIN, W.; YANG, X. An efficient framework for recognizing traffic lights in night traffic images. In: *2012 5th International Congress on Image and Signal Processing*. [S.l.: s.n.], 2012. p. 832–835. Citado na página 25.

FERNANDES, J. d. A.; MATOS, L. N.; ARAGÃO, M. G. d. S. Geometrical approaches for facial expression recognition using support vector machines. In: *2016 29th SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI)*. [S.l.: s.n.], 2016. p. 347–354. Citado 2 vezes nas páginas 14 e 23.

GEVARTER, W. *Artificial Intelligence, Expert Systems, Computer Vision, and Natural Language Processing*. Noyes Publications, 1984. ISBN 9780815509943. Disponível em: <<https://books.google.com.br/books?id=doFQAAAAMAAJ>>. Citado 2 vezes nas páginas 15 e 17.

GOMEZ, A. E. et al. Traffic lights detection and state estimation using hidden markov models. In: *2014 IEEE Intelligent Vehicles Symposium Proceedings*. [S.l.: s.n.], 2014. p. 750–755. ISSN 1931-0587. Citado na página 25.

GONG, J. et al. The recognition and tracking of traffic lights based on color segmentation and camshift for intelligent vehicles. In: *2010 IEEE Intelligent Vehicles Symposium*. [S.l.: s.n.], 2010. p. 431–435. ISSN 1931-0587. Citado na página 24.

- GONZALEZ R. C. E WOODS, R. E. *Processamento Digital de Imagens - 3ª ed.* [S.l.]: São Paulo: Pearson, 2010. Citado 3 vezes nas páginas 14, 15 e 16.
- Gonçalves, Jéssica. *Desrespeito à sinalização causa 7,5 mil acidentes no trânsito em 2014*. 2015. Acesso em: 09 ago. 2017. Disponível em: <<http://www.ebc.com.br/noticias/2015/07/desrespeito-sinalizacao-causa-75-mil-acidentes-no-transito-em-2014>>. Citado na página 10.
- GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. *Deep Learning*. [S.l.]: MIT Press, 2016. <<http://www.deeplearningbook.org>>. Citado 3 vezes nas páginas 19, 21 e 22.
- GRANDO, N. Segmentação de imagens tomográficas visando a construção de modelos médicos. Dissertação de Mestrado do Programa de Pós-Graduação em Engenharia Elétrica e Informática Industrial. CEFET-PR - Centro Federal de Educação Tecnológica do Paraná, 2005. Citado 2 vezes nas páginas 15 e 18.
- HAIR, J. et al. Multivariate data analysis 6th edition prentice hall. *New Jersey*, 2006. Citado na página 25.
- HAYKIN, S. S. et al. *Neural networks and learning machines*. [S.l.]: Pearson Upper Saddle River, 2009. v. 3. Citado 4 vezes nas páginas 21, 22, 23 e 28.
- JANG, C. et al. Multiple exposure images based traffic light recognition. In: *2014 IEEE Intelligent Vehicles Symposium Proceedings*. [S.l.: s.n.], 2014. p. 1313–1318. ISSN 1931-0587. Citado na página 25.
- JAYARAMAN, S. *DIGITAL IMAGE PROCESSING*. Tata McGraw-Hill Education, 2011. ISBN 9781259081439. Disponível em: <<https://books.google.com.br/books?id=ttNuj2mmt4sC>>. Citado na página 18.
- JENSEN, M. B. et al. Vision for looking at traffic lights: Issues, survey, and perspectives. *IEEE Transactions on Intelligent Transportation Systems*, v. 17, n. 7, p. 1800–1815, July 2016. ISSN 1524-9050. Citado 2 vezes nas páginas 10 e 11.
- Jl, Y. et al. Integrating visual selective attention model with hog features for traffic light detection and recognition. In: *2015 IEEE Intelligent Vehicles Symposium (IV)*. [S.l.: s.n.], 2015. p. 280–285. ISSN 1931-0587. Citado 2 vezes nas páginas 24 e 25.
- JIE, Y. et al. A new traffic light detection and recognition algorithm for electronic travel aid. In: *2013 Fourth International Conference on Intelligent Control and Information Processing (ICICIP)*. [S.l.: s.n.], 2013. p. 644–648. Citado na página 24.
- JOHN, V. et al. Saliency map generation by the convolutional neural network for real-time traffic light detection using template matching. *IEEE Transactions on Computational Imaging*, v. 1, n. 3, p. 159–173, Sept 2015. ISSN 2333-9403. Citado 2 vezes nas páginas 24 e 25.
- JOHN, V. et al. Traffic light recognition in varying illumination using deep learning and saliency map. In: *17th International IEEE Conference on Intelligent Transportation Systems (ITSC)*. [S.l.: s.n.], 2014. p. 2286–2291. ISSN 2153-0009. Citado 3 vezes nas páginas 24, 25 e 29.
- JOHN, V.; ZHEMING, L.; MITA, S. Robust traffic light and arrow detection using optimal camera parameters and gps-based priors. In: *2016 Asia-Pacific Conference on Intelligent Robot Systems (ACIRS)*. [S.l.: s.n.], 2016. p. 204–208. Citado na página 24.



- JÚNIOR, C. A. et al. Uma arquitetura híbrida lstm-cnn para reconhecimento de entidades nomeadas em textos naturais em língua portuguesa. *Encontro Nac. de Int. Artificial e Computacional - ENIAC (2016)*, 2016. Citado na página 22.
- Karpathy, A. *CS231n: Convolutional Neural Networks for Visual Recognition*. 2017. Acesso em: 04 ago. 2017. Disponível em: <<http://cs231n.github.io>>. Citado 2 vezes nas páginas 20 e 21.
- KIM, S. et al. Real time object detection based on saliency map. In: *2016 IEEE International Conference on Consumer Electronics (ICCE)*. [S.l.: s.n.], 2016. p. 534–535. Citado na página 24.
- LEVINSON, J. et al. Traffic light mapping, localization, and state detection for autonomous vehicles. In: *2011 IEEE International Conference on Robotics and Automation*. [S.l.: s.n.], 2011. p. 5784–5791. ISSN 1050-4729. Citado 2 vezes nas páginas 25 e 29.
- LI-TIAN, Z. et al. A framework of traffic lights detection, tracking and recognition based on motion models. In: *17th International IEEE Conference on Intelligent Transportation Systems (ITSC)*. [S.l.: s.n.], 2014. p. 2298–2303. ISSN 2153-0009. Citado na página 29.
- LORENA, A. C.; CARVALHO, A. C. P. L. F. de. Protein cellular localization with multiclass support vector machines and decision trees. In: *Proceedings of the 2005 Brazilian Conference on Advances in Bioinformatics and Computational Biology*. Berlin, Heidelberg: Springer-Verlag, 2005. (BSB'05), p. 42–53. ISBN 3-540-28008-1, 978-3-540-28008-8. Disponível em: <[http://dx.doi.org/10.1007/11532323\\_6](http://dx.doi.org/10.1007/11532323_6)>. Citado na página 23.
- LUGER, G. F. *Artificial intelligence: structures and strategies for complex problem solving*. [S.l.]: Pearson education, 2005. Citado na página 23.
- MASCETTI, S. et al. Robust traffic lights detection on mobile devices for pedestrians with visual impairment. *Comput. Vis. Image Underst.*, Elsevier Science Inc., New York, NY, USA, v. 148, n. C, p. 123–135, jul. 2016. ISSN 1077-3142. Disponível em: <<https://doi.org/10.1016/j.cviu.2015.11.017>>. Citado na página 25.
- MICHAEL, M.; SCHLIPSING, M. Extending traffic light recognition: Efficient classification of phase and pictogram. In: *2015 International Joint Conference on Neural Networks (IJCNN)*. [S.l.: s.n.], 2015. p. 1–8. ISSN 2161-4393. Citado na página 25.
- MICHALSKI, R. S.; CARBONELL, J. G.; MITCHELL, T. M. *Machine learning: An artificial intelligence approach*. [S.l.]: Springer Science & Business Media, 2013. Citado na página 19.
- MORALES, D.; CENTENO, T. M.; MORALES, R. C. Extração automática de marcadores anatômicos no desenvolvimento de um sistema de auxílio ao diagnóstico postural por imagens. III Workshop de Informática aplicada à Saúde - CBComp, 2003. Citado na página 14.
- MORGAN, J. Técnicas de segmentação de imagens na geração de programas para máquinas de comando numérico. Dissertação (Mestrado em Engenharia de Produção) - Universidade Federal de Santa Maria, 2008. Citado 2 vezes nas páginas 14 e 18.
- OMACHI, M.; OMACHI, S. Traffic light detection with color and edge information. In: *2009 2nd IEEE International Conference on Computer Science and Information Technology*. [S.l.: s.n.], 2009. p. 284–287. Citado na página 25.

- OMACHI, M.; OMACHI, S. Detection of traffic light using structural information. In: *IEEE 10th INTERNATIONAL CONFERENCE ON SIGNAL PROCESSING PROCEEDINGS*. [S.l.: s.n.], 2010. p. 809–812. ISSN 2164-5221. Citado 2 vezes nas páginas 24 e 25.
- ONIGA, F.; PRODAN, S.; NEDEVSKI, S. Traffic light detection on mobile devices. In: *2015 IEEE International Conference on Intelligent Computer Communication and Processing (ICCP)*. [S.l.: s.n.], 2015. p. 287–292. Citado na página 24.
- PEREIRA, E. T. Atenção visual *bottom-up* guiada por otimização via algoritmos genéticos. Campina Grande, 2007. Citado na página 17.
- PHILIPSEN, M. P. et al. Traffic light detection: A learning algorithm and evaluations on challenging dataset. In: *2015 IEEE 18th International Conference on Intelligent Transportation Systems*. [S.l.: s.n.], 2015. p. 2341–2345. ISSN 2153-0009. Citado na página 24.
- SALARIAN, M.; MANAVELLA, A.; ANSARI, R. A vision based system for traffic lights recognition. In: *2015 SAI Intelligent Systems Conference (IntelliSys)*. [S.l.: s.n.], 2015. p. 747–753. Citado na página 25.
- SANTOS, R. M. et al. Speech recognition in noisy environments with convolutional neural networks. In: *2015 Brazilian Conference on Intelligent Systems (BRACIS)*. [S.l.: s.n.], 2015. p. 175–179. Citado na página 22.
- SATHIYA, S.; BALASUBRAMANIAN, M.; PRIYA, D. V. Real time recognition of traffic light and their signal count-down timings. In: *International Conference on Information Communication and Embedded Systems (ICICES2014)*. [S.l.: s.n.], 2014. p. 1–6. Citado na página 24.
- SHI, X.; ZHAO, N.; XIA, Y. Detection and classification of traffic lights for automated setup of road surveillance systems. *Multimedia Tools Appl.*, Kluwer Academic Publishers, Hingham, MA, USA, v. 75, n. 20, p. 12547–12562, out. 2016. ISSN 1380-7501. Disponível em: <<http://dx.doi.org/10.1007/s11042-014-2343-1>>. Citado na página 25.
- SHI, Z.; ZOU, Z.; ZHANG, C. Real-time traffic light detection with adaptative background suppression filter. *IEEE Transactions on Intelligent Transportation Systems*, v. 17, n. 3, p. 690–700, March 2016. ISSN 1524-9050. Citado 17 vezes nas páginas 7, 8, 22, 24, 25, 26, 27, 29, 30, 31, 32, 37, 38, 45, 46, 49 e 50.
- SOOKSATRA, S.; KONDO, T. Red traffic light detection using fast radial symmetry transform. In: *2014 11th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON)*. [S.l.: s.n.], 2014. p. 1–6. Citado na página 24.
- SUNG, T.-P.; TSAI, H.-M. Real-time traffic light recognition on mobile devices with geometry-based filtering. In: *2013 Seventh International Conference on Distributed Smart Cameras (ICDSC)*. [S.l.: s.n.], 2013. p. 1–7. Citado na página 29.
- TREHARD, G. et al. Tracking both pose and status of a traffic light via an interacting multiple model filter. In: *17th International Conference on Information Fusion (FUSION)*. [S.l.: s.n.], 2014. p. 1–7. Citado na página 24.
- WANG, Z. et al. Traffic lights recognition based on pcanet. In: *2015 Chinese Automation Congress (CAC)*. [S.l.: s.n.], 2015. p. 559–564. Citado 2 vezes nas páginas 24 e 25.

- WEBER, M.; WOLF, P.; ZÖLLNER, J. M. Deeptlr: A single deep convolutional network for detection and classification of traffic lights. In: *2016 IEEE Intelligent Vehicles Symposium (IV)*. [S.l.: s.n.], 2016. p. 342–348. Citado 2 vezes nas páginas 24 e 25.
- WIDYANTORO, D. H.; SAPUTRA, K. I. Traffic lights detection and recognition based on color segmentation and circle hough transform. In: *2015 International Conference on Data and Software Engineering (ICoDSE)*. [S.l.: s.n.], 2015. p. 237–240. Citado na página 24.
- YING, J.; TIAN, J.; LEI, L. Traffic light detection based on similar shapes searching for visually impaired person. In: *2015 Sixth International Conference on Intelligent Control and Information Processing (ICICIP)*. [S.l.: s.n.], 2015. p. 376–380. Citado na página 29.
- YONEDA, K.; SUGANUMA, N.; ALDIBAJA, M. A. Simultaneous state recognition for multiple traffic signals on urban road. In: *2016 11th France-Japan 9th Europe-Asia Congress on Mechatronics (MECATRONICS) /17th International Conference on Research and Education in Mechatronics (REM)*. [S.l.: s.n.], 2016. p. 135–140. Citado na página 29.
- ZHANG, Y. et al. A multi-feature fusion based traffic light recognition algorithm for intelligent vehicles. In: *Proceedings of the 33rd Chinese Control Conference*. [S.l.: s.n.], 2014. p. 4924–4929. Citado 2 vezes nas páginas 24 e 29.
- ZHOU, Y.; CHEN, Z.; HUANG, X. A system-on-chip fpga design for real-time traffic signal recognition system. In: *2016 IEEE International Symposium on Circuits and Systems (ISCAS)*. [S.l.: s.n.], 2016. p. 1778–1781. Citado 2 vezes nas páginas 24 e 25.
- ZONG, W.; CHEN, Q. Traffic light detection based on multi-feature segmentation and online selecting scheme. In: *2014 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*. [S.l.: s.n.], 2014. p. 204–209. ISSN 1062-922X. Citado na página 24.